VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Kryptoanalýza klasických šifrovacích algoritmů

Cryptanalysis of Classical Cryptographic Algorithms

2019 Josef Procházka

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Josef Procházka**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 1801T064 Informační a komunikační bezpečnost

Téma: Kryptoanalýza klasických šifrovacích algoritmů
Cryptanalysis of Classical Cryptographic Algorithms

Jazyk vypracování: čeština

Zásady pro vypracování:

Kryptoanalýza se zabývá studiem kryptografických systémů s cílem pochopit, jak fungují, a zda mají nějaké nedostatky, které by umožnily jejich napadení či prolomení. Kryptoanalytické metody se vyvíjejí, od jednoduchých aplikovatelných na jednoduché kryptografické systémy až po moderní složitější metody používané pro kryptoanalýzu současných kryptografických systémů. Cílem diplomové práce je implementace některých jednoduchých kryptoanalytických metod.

1. Seznamte se s typy kryptoanalytických útoků, zejména s útokem ze známého šifrového textu.
2. Seznamte se s kryptoanalytickými metodami používanými pro klasické (historické) šifrovací algoritmy.
3. Navrhněte a naimplementujte aplikaci pro kryptoanalýzu vybranými metodami.
4. Ověřte funkčnost své implementace, vhodně zdokumentujte.

Seznam doporučené odborné literatury:

[1] Sinkov A.: Elementary Cryptanalysis ISBN-10: 0883856476
[2] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone: Handbook of Applied Cryptography, ISBN: 0-8493-8523-7
[3] Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **RNDr. Eliška Ochodková, Ph.D.**

Datum zadání: 01.09.2017
Datum odevzdání: 30.04.2019

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. června 2019 Procházka .....

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 28. června 2019 ..... Procházka

Chtěl bych poděkovat své vedoucí, RNDr. Elišce Ochodkové, Ph.D., za pomoc a trpělivost při tvorbě této práce.

Abstrakt

Cílem diplomové práce je návrh a implementace aplikace, která bude schopna šifrování a dešifrování vybranými historickými metodami a dále kryptoanalýzy šifrových textů, vytvořených danými metodami. První část se věnuje známým šifrám a jejich principům. Druhá část popisuje kryptoanalytické metody pro substituční a transpoziční šifry. Poslední část již pojednává o vytvořené aplikaci, kde popisuje její implementaci, ovládání a funkčnost.

Klíčová slova: Kryptografie, Kryptoanalýza, Útok ze známého šifrového textu, C#

Abstract

The aim of this thesis is to design and implement an application that will be able to encrypt and decrypt texts using chosen historical methods and also to perform cryptoanalysis of the cipher texts created by these methods. The first part of this thesis is focused on known ciphers and their principles. The second part describes cryptoanalytical methods for substitution and transposition ciphers. The last part talks about the application itself, describing its implementation, usage and functionality.

Key Words: Cryptography, Cryptanalysis, Ciphertext-only attack, C#

Obsah

Seznam použitých zkratek a symbolů **15**

Seznam obrázků **17**

Seznam tabulek **19**

1 Úvod **21**

2 Terminologie **23**
2.1 Základní pojmy 23
2.2 Symboly 24

3 Klasické (historické) šifry **27**
3.1 Substituční šifry 27
3.2 Transpoziční šifry 38

4 Kryptoanalýza klasických šifer **45**
4.1 Frekvenční analýza 45
4.2 Ohodnocující funkce (scoring functions) 47
4.3 Určení pravděpodobné šifry 51
4.4 Vzdálenost jednoznačnosti (unicity distance) 52
4.5 Útok hrubou silou (brute-force attack) 53
4.6 Slovníkový útok 54
4.7 Metaheuristické metody 54

5 Návrh a analýza aplikace **61**
5.1 Implementace 61
5.2 Šifrování/Dešifrování 62
5.3 Analýza šifrového textu 64
5.4 Kryptoanalýza 67
5.5 Kryptoanalýza Caesarovy šifry 67
5.6 Kryptoanalýza Afinní šifry 69
5.7 Kryptoanalýza Vigenèrovy šifry 70
5.8 Kryptoanalýza transpoziční šifry 71
5.9 Kryptoanalýza pomocí "shotgun restart" horolezeckého algoritmu 73
5.10 Kryptoanalýza obecné substituční šifry (manuálně) 74

6 Závěr **79**

Literatura **81**

Přílohy **81**

A Třídní diagram **83**

Seznam použitých zkratek a symbolů

NSD	–	Největší společný dělitel
PC	–	Personal computer

Seznam obrázků

1 Polybiův čtverec s číslicemi a písmeny 30
2 Albertiho šifrovací disk [10] 32
3 Vigenèrův čtverec [11] 33
4 Šifrování Playfairovou šifrou v případě, že je dvojice písmen: 1. na jednom řádku 2. v jednom sloupci 3. v odišném řádku a sloupci 36
5 Histogram četností anglických písmen [12] 45
6 Histogram četností písmen z náhodných slov (vytvořeno v programu Kryptoanalýza) 46
7 Histogram četností písmen z náhodných slov šifrovaných Caesarovou šifrou (vytvořeno v programu Kryptoanalýza) 46
8 Histogram písmen z náhodných slov šifrovaných Vigenèrovou šifrou (vytvořeno v programu Kryptoanalýza) 47
9 Chí-kvadrát test aplikovaný na rozluštění Caesarovy šifry (vytvořeno v programu Kryptoanalýza) 49
10 Výsledky fitness funkce použité na rozluštění Caesarovy šifry (vytvořeno v programu Kryptoanalýza) 51
11 Příklad ideálního prostoru k prohledávání gradientním algoritmem [1] 58
12 Příklad hladkého prostoru k prohledávání gradientním algoritmem [1] 59
13 Příklad hrbolatého prostoru k prohledávání gradientním algoritmem [1] 59
14 Grafické rozhraní aplikace Kryptoanalýza 61
15 Menu položky *Šifrování/Dešifrování* v aplikaci Kryptoanalýza 63
16 Zadání klíče v aplikaci Kryptoanalýza 63
17 Zadání klíče pro Caesarovu šifru v aplikaci Kryptoanalýza 64
18 Menu položky *Analýza* v aplikaci Kryptoanalýza 64
19 Zobrazení histogramu pro připravený text v aplikaci Kryptoanalýza 65
20 Formulář pro zobrazení n-gramů v aplikaci Kryptoanalýza 65
21 Textové okno s n-gramy v aplikaci Kryptoanalýza 66
22 Formulář pro určení šifry (Vigenèrova) v aplikaci Kryptoanalýza 66
23 Formulář pro určení šifry (Transpoziční) v aplikaci Kryptoanalýza 66
24 Menu položky *Kryptoanalýza* v aplikaci Kryptoanalýza 67
25 Porovnání dvou histogramů při kryptoanalýze textu šifrovaného Caesarovou šifrou 68
26 Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Caesarovu šifru (Chí-kvadrát) 68
27 Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Caesarovu šifru (Fitness funkce) 69
28 Porovnání dvou histogramů při kryptoanalýze textu šifrovaného Afinní šifrou . . 69
29 Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Afinní šifru (Chí-kvadrát) 70

30 Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Afinní šifru (Fitness funkce) . . . 70
31 Porovnání dvou histogramů při kryptoanalýze textu šifrovaného Vigenèrovou šifrou 71
32 Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Vigenèrovu šifru 71
33 Formulář pro kryptoanalýzu transpoziční šifrou 72
34 Porovnání dvou histogramů při kryptoanalýze textu šifrovaného Transpoziční šifrou 72
35 Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Transpoziční šifru . . . 73
36 Formulář pro kryptoanalýzu obecné substituční šifry 73
37 Výsledky kryptoanalýzy v tabulce pro Obecnou substituční šifru 74
38 Histogram šifrového textu pro manuální kryptoanalýzu 75
39 Formulář pro manuální kryptoanalýzu 76
40 Vložení písmena P do prvního okénka ve formuláři pro manuální kryptoanalýzu . 76
41 Příklad nahrazení písmena šifrového textu potencionálními písmeny otevřeného textu . . . 76
42 Nalezení trigramu *TfE* (*THE*) 77
43 Nalezení slova *THEx* (*THEY*) 77
44 Nalezení slova *HEq* (*HER*) 77
45 Nalezení slova *nTHER* (*OTHER*) 77
46 Třídní diagram aplikace 84

Seznam tabulek

1 Šifrování textu Caesarovou šifrou s klíčem 3 27
2 Šifrování textu pomocí afinní šifry 28
3 Dešifrování textu pomocí afinní šifry 29
4 Permutace abecedy 29
5 Šifrování textu Substituční šifrou 29
6 Šifrování pomocí Polybiova čtverce 30
7 Otevřený text, klíč a šifrový text při šifrování Vernamovou šifrou 31
8 Otevřený text a klíč u šifrování Vigenèrovou šifrou 34
9 Otevřený text, klíč a šifrový text při šifrování Vigenèrovou šifrou 34
10 Obecný klíč pro Playfairovu šifru 35
11 Klíč pro Playfairovu šifru s klíčovým slovem *KEY* 37
12 Šifrování Rail Fence šifrou 39
13 Příprava tabulky pro dešifrování Rail Fence šifrou 39
14 Příprava tabulky pro šifrování sloupcovou transpozicí s úplnou tabulkou 40
15 Tabulka po permutaci sloupců 40
16 Transponování tabulky pro získání otevřeného textu 41
17 Naplnění tabulky šifrovým textem 42
18 Tabulka naplněná otevřeným textem před transpozicí 42
19 Tabulka naplněná otevřeným textem po transpozici 42
20 Transponování tabulky pro získání otevřeného textu 43
21 Vkládání šifrového textu do tabulky 43
22 Výsledky výpočtů IK pro délky klíče použitého u šifrování Vigenèrovou šifrou . . 48
23 Počet možných klíčů transpoziční šifry některých délek klíče [6] 53

1 Úvod

Již několik tisíc let mají mnozí lidé na tomto světě potřebu některé informace uchovávat v tajnosti, šifrovat je a zkrátka zamezovat tomu, aby je znal, nebo o nich vůbec věděl, někdo cizí. Tuto skupinu lidí časem logicky následovala další skupina mající snahu se k těmto utajeným informacím dostat za pomoci metod, které se časem vyvíjely a stále vyvíjejí s příchodem moderních informačních systémů. Tato druhá skupina lidí časem položila základy nového vědního oboru zvaného kryptoanalýza.

Kdysi stačila k rozluštění tajných zpráv pouhá tužka, papír a lidský rozum, což začalo být s nástupem moderních technologií zcela nedostatečné. Především 20. století přineslo velký růst a větší rozmanitost v kryptoanalýze. S příchodem výkonnějších výpočetních systémů bylo dovoleno nejen tvořit mnohem složitější šifry, ale také využívat tyto výkonné prostředky k jejich analýze a rozluštění. Kryptoanalýza měla své uplatnění i mezi lidmi v běžném životě. Posílali si šifrované zprávy, jejichž obsahu se jiní chtěli zmocnit. Byla ale také velmi důležitá v období válek, kdy rozluštění šifrované zprávy nepřítele mohlo zachránit mnoho životů a umožňovalo provést nečekaný útok.

Cílem této práce je prozkoumat kryptoanalytické metody používané pro klasické (historické) šifrovací algoritmy. Jsou zde popsány jak tyto algoritmy a jejich šifrovací i dešifrovací systém, tak i metody, kterými lze šifrové texty rozluštit. V práci je rozebráno především lámání šifer pro případy, kdy je znám pouze šifrový text bez použitého klíče. Součástí je i praktické vytvoření aplikace provádějící tyto klasické šifrovací a dešifrovací algoritmy a také poskytuje kryptoanalýzu vybranými metodami pro některé šifry. Funkčnost této aplikace je zdokumentovaná v poslední kapitole. Aplikace je napsaná v programovacím jazyce C#.

První kapitola slouží jako úvod do světa kryptologie. Je zaměřená na obecné pojmy, základní kryptologické informace a historické šifrovací algoritmy. Ve druhé a třetí kapitole se již věnujeme konkrétním substitučním a transpozičním šifrám. To na mnoha místech zahrnuje i popis několika kryptoanalytických metod, které lze u daných šifer použít. Čtvrtá kapitola obsahuje obecné informace o konkrétních kryptoanalytických metodách. Popíšeme zde podrobněji jejich principy a využití. V páté kapitole je popsána vytvořená aplikace, její přínos a celkové zhodnocení její funkčnosti. Naším cílem je seznámit čtenáře s pojmem *kryptoanalýza* a prakticky ukázat využití pro historické šifry.

2 Terminologie

2.1 Základní pojmy

Kryptografie a k ní patřící nauka o šifrovacích algoritmech má vliv na naši historii již více než 2000 let. V rámci této vědní disciplíny vznikl nemalý počet pojmů a termínů. Ty je nezbytné znát pro správné vysvětlení a pochopení šifer, kterým se v práci věnujeme, a také pro pochopení kryptoanalýzy obecně. Proto je na začátku kapitoly dán prostor k vysvětlení těchto základních pojmů.

Kryptologie - věda věnující se šifrování a dešifrování informací. Skládá se z kryptografie a kryptoanalýzy [2].

Kryptografie - zabývá se konstrukcí šifrovacích algoritmů, tedy nástrojů napomáhajících k šifrování zpráv. Jedná se o matematické metody zajišťující důvěrnost zprávy, autentizaci entit (ověření subjektu) a integritu dat (neporušitelnost) [2].

Kryptoanalýza - věda, která se soustředí na vlastnosti otevřeného textu, šifrovacích klíčů a šifrového textu. Prakticky se jedná o snahu o prolomení skryté informace a zkoumání odolnosti kryptografického systému [2].

Kryptoanalytik - člověk zabývající se kryptoanalýzou [2].

Kryptoanalytická metoda - typ útoku použitého k získání šifrového klíče nebo k prolomení šifry [2].

Kryptoanalytický útok - snaha o odhalení slabých míst šifry a jejich zneužití. Tyto útoky se zakládají na matematické analýze šifry [2].

Steganografie - slovo pochází z řeckého jazyka (steganós - ukrytý, gráphein - psát) a vyjadřuje ukrývání informací nebo zpráv takovým způsobem, aby nepovolané osoby nevěděly o jejich existenci. Steganografie proto není šifrování zpráv jako takové [2].

Přípustná abeceda - konečná množina znaků, z níž se vybírají prvky tvořící otevřený i šifrový text. Patří sem otevřená abeceda tvořená ze znaků pro otevřený text a šifrová abeceda tvořená ze znaků pro šifrový text [2].

Otevřený text - v kryptografii se tak nazývá běžný čitelný text v nějakém jazyce. V kryptologii je otevřený text tím, co se nesmí dostat do rukou útočníka, a proto se tato informace šifruje. Abecedou neboli znaky v tomto textu jsou míněna jakákoliv písmena, interpunkce nebo číslice. Množinu otevřených textů označujeme symbolem M. [2]

Šifrový text - výsledný text, který vznikne šifrováním otevřeného textu. Někdy se také nazývá kryptogram. Je nečitelný - skládá se z prvků šifrové abecedy. Množina šifrových textů se značí symbolem C [2].

Šifrování - proces zakódování (zamaskování) otevřeného textu. Při tomto procesu je otevřený text převeden podle algoritmu a šifrovacího klíče na šifrový text [2]. Symbolem E značíme množinu šifrovacích funkcí (algoritmů) [2].

Dešifrování - získání otevřeného textu ze šifrového textu zpravidla postupnými operacemi opačnými k šifrování. Člověk nebo entita provádějící dešifrování, přitom zná použitý šifrovací systém a šifrovací klíč [2]. Symbolem D značíme množinu dešifrovacích funkcí (algoritmů) [2].

Šifrovací klíč - informace ovlivňující šifrování otevřeného textu. Určuje do jaké podoby bude informace šifrována. Nejčastěji to bývá množina čísel, znaků, nebo jejich kombinací. Symbolem K označujeme konečnou množinu možných klíčů [2].

Dešifrovací klíč - informace ovlivňující dešifrování šifrového textu. Dešifrovací klíč může být totožný s šifrovacím klíčem [2].

Útok ze známého šifrového textu (Ciphertext-only attack) - způsob kryptoanalytického útoku, při němž se útočník snaží zjistit klíč nebo otevřený text pouze pomocí zkoumání šifrového textu [2].

Útok ze známého otevřeného textu (Known-plaintext attack) - je způsob kryptoanalytického útoku, při němž má útočník k dispozici otevřený text a k němu odpovídající šifrový text. V tomto případě je snaha získat klíč, kterým byl daný text šifrován [2].

Útok hrubou silou (Brute force attack) - druh útoku, kdy útočník postupně zkouší všechny možné klíče k rozluštění šifrového textu. Pro každý klíč tak získá jeden otevřený text a zjišťuje, jestli z něj může dostat nějaké informace, respektive zda získal původní otevřený text [2].

Frekvenční analýza - je útočníkovo využití znalosti jazyka a jeho statistických vlastností. Tyto znalosti využívá k získání informací vedoucí k rozluštění šifrového textu [2].

2.2 Symboly

Zde je podrobněji popsán seznam symbolů používaných v této práci:

- A označuje přípustnou abecedu. Například $A = \{0, 1\}$ označuje binární abecedu [2]. V této práci bude A ve většině případů označovat anglickou abecedu.
- M označuje prostor otevřených textů obsahující řetězce či znaky přípustné abecedy. Prvek množiny M se nazývá otevřený text [2].
- C označuje prostor šifrových textů obsahující řetězce či znaky přípustné abecedy, který se může lišit od přípustné abecedy pro M. Prvek množiny C se nazývá šifrový text [2].
- K označuje prostor klíčů. Prvek množiny K se nazývá klíč [2].

- Každý prvek $e \in K$ jednoznačně určuje bijekci z C do M, která se značí E_e. E_e se nazývá šifrovací funkce nebo šifrovací transformace. [2].
- Pro každé $d \in K$ platí, že D_d označuje bijekci z C do M. D_d se nazývá dešifrovací funkce nebo dešifrovací transformace [2].
- Proces aplikování transformace E_e na otevřený text $m \in M$ obvykle odkazuje na šifrování m [2].
- Proces aplikování transformace D_d na šifrový text c obvykle odkazuje na dešifrování c [2].

3 Klasické (historické) šifry

3.1 Substituční šifry

3.1.1 Caesarova šifra

Caesarova šifra je jednou z nejstarších substitučních šifer a často bývá tou první, se kterou se lidé setkají, když se začnou o kryptografii zajímat [5]. Obecně je princip šifrování Caesarovou šifrou založen na tom, že se postupně každé písmeno otevřeného textu zamění za písmeno nacházející se v abecedě o tři pozice dopředu.

Caesarova šifra je jednoduchá substituční šifra s permutací e vytvořenou posunem každého písmena v abecedě o hodnotu k. Přesněji řečeno, jestliže $|A| = s$ a m_i obsahuje celočíselné hodnoty $i, 0 \leq i \leq s-1$, potom platí:

$$c_i = e(m_i) = m_i + k \bmod s, \tag{1}$$

kde $s = 26$ pro anglickou abecedu, $k = 3$ a písmena od A do Z se dosazují jako celá čísla od 0 po 25 [2]. To znamená, že se při šifrování písmeno A zamění za písmeno **D**, písmeno B za písmeno **E** atd. V tabulce 1 je ukázáno šifrování slova *SOLO*.

Tabulka 1: Šifrování textu Caesarovou šifrou s klíčem 3

Otevřený text:	SOLO
Šifrový text:	**VROR**

Zde si můžeme povšimnout, že obě písmena O se v předchozím slově šifrovala na totožná písmena, což je jednou ze slabin Caesarovy šifry a jiných substitučních šifer. Pro dešifrování platí tato rovnice:

$$m_i = d(c_i) = c_i - k \bmod s. \tag{2}$$

Je také důležité podotknout, jak se šifrují poslední tři písmena abecedy, u nichž zajišťuje správný posun právě zmíněné aritmetické modulo. Při šifrování každého písmena se provádí posun v abecedě o hodnotu klíče a pokud se v průběhu posunu dojde až na konec abecedy, bude posun pokračovat opět od začátku abecedy. Pokud má klíč hodnotu 3, trojice písmen *XYZ* se šifruje na **ABC**.

Podobného výsledku lze dosáhnout jiným posunem za použití čísla $k = 1, 2, \ldots, 25$. V takových případech se tato šifra obecně nazývá *Shift šifra*. U této šifry se provádí šifrování a dešifrování také podle zmíněných rovnic 1 a 2. Vzhledem k tomu, že Shift šifra má pouze 25 různých možných klíčů, je k jejímu prolomení vhodný útok hrubou silou.

3.1.2 Afinní šifra

Afinní šifra poskytuje oproti Caesarově šifře větší rozmezí klíčů. Klíč je složen celkově ze dvou čísel, respektive ze tří s tím, že třetí číslo bývá obvykle *26* a jedná se o počet písmen v otevřené abecedě. Princip této substituční šifry je v tom, že každé písmeno se převede na číslo, které je potom šifrováno pomocí jednoduché matematické funkce a poté převedeno zpět na písmeno.

Zmíněná dvě čísla v tomto klíči označujeme písmeny a a b. Třetí číslo se označuje písmenem m. Písmena otevřeného textu převedeme na čísla stejným způsobem, jako v kapitole 3.1.1 [6].

Čísla a a b musejí být nesoudělná čísla. Za a dosazujeme čísla o hodnotách 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25. Co se týče čísla b, zde můžeme dosadit jakékoliv číslo od 0 do 25 [6]. Číslo m má v našich příkladech vždy hodnotu 26. Tato pravidla zajišťují možnost správného dešifrování šifrového textu. V případě že A obsahuje anglickou abecedu, vzorec pro šifrování vypadá takto:

$$e_K(m) = am + b \bmod 26, \tag{3}$$

kde $0 \leq a, b \leq 25$. Klíč je (a, b). Znak m je písmeno otevřeného textu [2]. Šifrový text $c = e_K(x)$ je dešifrován použitím rovnice

$$d_K(c) = (c - b)\, a^{-1} \bmod 26, \tag{4}$$

s nezbytnou podmínkou pro získání inverzního prvku, kde platí $NSD(a, 26) = 1$. c je znak šifrového textu a a^{-1} je inverzní prvek k prvku a, vypočítaný Rozšířeným Euklidovým algoritmem. Pro konkrétní příklad si zvolme klíč, jehož hodnoty jsou $a = 5$, $b = 9$ a $m = 26$. Náš otevřený text je *AFFINE*. Nyní začneme šifrovat postupně každé písmeno v otevřeném textu.

Tabulka 2: Šifrování textu pomocí afinní šifry

A	$\Rightarrow$	0	$\Rightarrow$	$(5 \cdot 0 + 9) \bmod 26$	$\Rightarrow$	9	$\Rightarrow$	**J**
F	$\Rightarrow$	5	$\Rightarrow$	$(5 \cdot 5 + 9) \bmod 26$	$\Rightarrow$	8	$\Rightarrow$	**I**
F	$\Rightarrow$	5	$\Rightarrow$	$(5 \cdot 5 + 9) \bmod 26$	$\Rightarrow$	8	$\Rightarrow$	**I**
I	$\Rightarrow$	8	$\Rightarrow$	$(5 \cdot 8 + 9) \bmod 26$	$\Rightarrow$	23	$\Rightarrow$	**X**
N	$\Rightarrow$	13	$\Rightarrow$	$(5 \cdot 13 + 9) \bmod 26$	$\Rightarrow$	22	$\Rightarrow$	**W**
E	$\Rightarrow$	4	$\Rightarrow$	$(5 \cdot 4 + 9) \bmod 26$	$\Rightarrow$	3	$\Rightarrow$	**D**

Výsledný šifrový text má podobu **JIIXWD**. Můžeme jasně vidět, že písmeno F se šifrovalo dvakrát ze dvou případů na stejné písmeno. Tento šifrový text si nyní dešifrujeme vzorcem pro dešifrování. V tomto případě platí, že $a^{-1} = 21$, což plyne z Rozšířeného Euklidového algoritmu.

Tabulka 3: Dešifrování textu pomocí afinní šifry

J	⇒	9	⇒	$(9-9)\cdot 21 \bmod 26$	⇒	0	⇒	**A**
I	⇒	8	⇒	$(8-9)\cdot 21 \bmod 26$	⇒	5	⇒	**F**
I	⇒	8	⇒	$(8-9)\cdot 21 \bmod 26$	⇒	5	⇒	**F**
X	⇒	23	⇒	$(23-9)\cdot 21 \bmod 26$	⇒	8	⇒	**I**
W	⇒	22	⇒	$(22-9)\cdot 21 \bmod 26$	⇒	13	⇒	**N**
D	⇒	3	⇒	$(3-9)\cdot 21 \bmod 26$	⇒	4	⇒	**E**

3.1.3 Obecná substituce

U této šifry dochází při šifrování také k nahrazování písmen z jedné množiny za písmena z druhé množiny. Tou první je otevřená abeceda tvořící písmena anglické abecedy tak, jak jdou po sobě. Šifrová abeceda obsahuje permutaci těchto písmen anglické abecedy.

Nechť A je abeceda znaků a M je množina všech řetězců délky t nad abecedou A. Nechť K je množina všech permutací A. Pro každé $e \in K$ platí šifrovací transformace E_e:

$$E_e(m) = (e(m_1)e(m_2)\cdots e(m_t)) = (c_1c_2\cdots c_t) = c, \tag{5}$$

kde $m = (m_1m_2\cdots m_t) \in M$. Jinými slovy, každé písmeno v t-tici se nahradí (substituuje) jiným písmenem z množiny A v závislosti na pevně dané permutaci e. Pro dešifrování potom platí:

$$D_d(c) = (d(c_1)d(c_2)\cdots d(c_t)) = (m_1m_2\cdots m_t) = m. \tag{6}$$

E_e se nazývá jako *jednoduchá substituční šifra*, nebo také *monoalfabetická substituční šifra* [2]. V následující tabulce 4 je příklad otevřené abecedy a šifrové abecedy pod sebou ve dvou řádcích.

Tabulka 4: Permutace abecedy

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
S	**O**	**Q**	**Z**	**N**	**B**	**A**	**Y**	**U**	**T**	**R**	**V**	**P**	**E**	**F**	**M**	**X**	**K**	**G**	**J**	**W**	**L**	**I**	**C**	**H**	**D**

Tabulka 4 je jedním z klíčů, přesněji řečeno - klíčem je šifrová abeceda ve spodním řádku. Otevřená abeceda zůstává vždy ve stejném pořadí. Princip šifrování je následující - každé písmeno, které chceme zašifrovat, vždy nalezneme v otevřené abecedě, což jsou v tomto případě písmena nacházející se nahoře. Takové písmeno nahradíme písmenem, které se nachází dole, což jsou v tomto případě písmena psané tučně. U dešifrování se provádí opačný postup. V tabulce 5 je ukázáno šifrování slova *JUDGE*.

Tabulka 5: Šifrování textu Substituční šifrou

Otevřený text:	JUDGE
Šifrový text:	**TWZAN**

Počet všech možných permutací anglické abecedy je roven číslu 26!, což je přibližně $4 \cdot 10^{26}$. Počet možných klíčů naznačuje, že v tomto případě již útok hrubou silou není tak rychle proveditelný jako u Caesarovy šifry.

3.1.4 Polybiův čtverec

Polybios byl starověký řecký politik narozený okolo roku 200 př. n. l. I přes to, že byl politikem a historikem, popsal v jednom ze svých děl jednoduchý a pro tu dobu velmi efektivní způsob, jak šifrovat písmena ve zprávě. To se později stalo užitečným nástrojem v telegrafii [8]. Jedná se o *Polybiův čtverec*. Jeho zobrazení pro anglickou abecedu můžeme vidět na obrázku 1.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I	J
3	K	L	M	N	O
4	P	Q	R	S	T
5	V	W	X	Y	Z

	A	B	C	D	E
A	A	B	C	D	E
B	F	G	H	I	J
C	K	L	M	N	O
D	P	Q	R	S	T
E	V	W	X	Y	Z

Obrázek 1: Polybiův čtverec s číslicemi a písmeny

Do čtverce je potřeba umístit všechna písmena, jenž tvoří otevřenou abecedu. V tomto případě bylo vynecháno písmeno *U*, jelikož se dá chápat jako *V*. Ve čtverci je prostor pouze pro 25 písmen.

Princip šifrování pomocí levé tabulky je takový, že vezmeme postupně každé písmeno otevřeného textu (v případě písmena *U* bereme v úvahu písmeno *V*) a toto písmeno nahradíme dvěma čísly. První zpravidla označuje číslo řádku, na němž se písmeno nachází, a druhé číslo sloupce. Písmeno *A* bychom potom nahradili číslicemi **11**, písmeno *B* potom číslicemi **12** atd. Dešifrování probíhá opačně. V šifrovacím textu, kde máme pouze čísla, bychom každou dvojici nahradili jedním písmenem podle této tabulky. V tabulce číslo 6 je ukázka šifrování slova *POLYBIUS* pomocí levé tabulky na číslice a pomocí pravé tabulky na písmena.

Tabulka 6: Šifrování pomocí Polybiova čtverce

Otevřený text:	POLYBIUS
Šifrový text tvořený z číslic:	**4135325412245144**
Šifrový text tvořený z písmen:	**DACECBEDABBDEADD**

3.1.5 Vernamova šifra

Tato šifra je definována na abecedě $A = \{0, 1\}$. Otevřený text $m_1 m_2 \cdots m_t$ v binární podobě je šifrován v závislosti na klíči v binární podobě $k_1 k_2 \cdots k_t$ stejné délky k vytvoření binární podoby

šifrového textu $c_1c_2 \cdots c_t$, jako:

$$c_i = m_i \oplus k_i, 1 \leq i \leq t. \tag{7}$$

Pokud je klíč náhodně vytvářen bez opakovaného použití, Vernamova šifra se nazývá *One-time pad*. Obdobně pak rovnice pro dešifrování vypadá takto:

$$m_i = c_i \oplus k_i, 1 \leq i \leq t \tag{8}$$

[2]. Pro příklad si zašifrujeme slovo VERNAM. Písmena otevřeného textu nepřevádíme do binární podoby, ale na decimální čísla. Princip je v tom, že každé písmeno otevřeného textu posuneme o náhodně zvolený počet míst v abecedě. Zvolíme proto náhodný klíč *BIGJOY* o stejné délce obsahující čísla 7, 11, 3, 9, 14, 24.

Tabulka 7: Otevřený text, klíč a šifrový text při šifrování Vernamovou šifrou

Otevřený text:	V	E	R	N	A	M
Klíč:	B	I	G	J	O	Y
Šifrový text:	**W**	**M**	**X**	**W**	**O**	**K**

Pro tuto šifru jsou dány tři podmínky spolehlivosti:

1. Klíč je stejně dlouhý jako zpráva otevřeného textu.
2. Písmena/čísla klíče jsou dokonale náhodné.
3. Stejný klíč se nepoužije opakovaně pro další šifrování [5].

Bez znalosti klíče je nemožné zprávu rozluštit. Kdybychom v tomto případě vyzkoušeli všechny možné klíče, mezi dešifrovanými texty bychom nalezli všechna slova, jenž lze vytvořit ze šesti písmen. Nemohli bychom usoudit, které z nich bylo v otevřeném textu.

3.1.6 Albertiho šifrovací disk

V 15. století začaly vznikat polyalfabetické šifry - za prvního objevitele této šifry je pokládán Leon Battista Alberti [9]. Polyalfabetická šifra se od běžné monoalfabetické šifry liší ve způsobu šifrování. U polyalfabetických šifer existují vždy minimálně dvě šifrové abecedy. V praxi to znamená, že u polyalfabetických šifer může být na jednom místě písmeno D nahrazeno určitým písmenem nebo posloupností písmen a na jiném místě ve stejném textu může být nahrazeno zcela jiným písmenem.

Množina klíčů K obsahuje všechny množiny permutací $(p_1, p_2, \cdots p_t)$, kde t značí délku klíče. Každá permutace p_i je definována na otevřené abecedě A. Šifrování zprávy $m = (m_1, m_2, \cdots m_t)$ podle klíče $e = (p_1, p_2, \cdots p_t)$ je potom definováno jako:

$$E_e(m) = (p_1(m_1)p_2(m_2) \cdots p_t(m_t)) \tag{9}$$

a dešifrování podle klíče sdruženého s $e = (p_1, p_2, \cdots p_t)$ je $d = (p_1^{-1}, p_2^{-1}, \cdots, p_t^{-1})$, kde p^{-1} značí inverzní permutaci [2].

Způsob jak provést šifrování polyalfabetickou šifrou lze provést například podle tzv. Albertiho šifrovacího disku, jenž je na obrázku 2.

Obrázek 2: Albertiho šifrovací disk [10]

Můžeme brát v úvahu, že vnější kruh zobrazuje písmena, které bychom hledali v našem otevřeném textu. Dané písmeno potom nahrazujeme jedním jiným písmenem z vnitřního kruhu ležícího pod ním. Pokud v určitých chvílích vnitřním kruhem pootočíme, dostáváme mnohem silnější polyalfabetickou šifru. Většinou se do šifrového textu psala písmena malá a pokud šifrový text obsahoval náhle velké písmeno, bylo to znakem toho, že při šifrování došlo k pootočení vnitřního kruhu, čímž se následně změnila celá šifrová abeceda.

Vznik polyalfabetických šifer byl následek příchodu frekvenční analýzy, jejíž použití bylo tímto způsobem v určitém smyslu znemožněno. Celkový šifrový text tímto totiž ztratil typické rysy, které se v něm obvykle objevovaly. Tím zmizela i určitá vodítka vedoucí k prolomení těchto šifer. I přesto se ale později objevili lidé jako byl Charles Babbage, kterým se podařilo i tyto typy šifer rozluštit.

3.1.7 Vigenèrova šifra

V minulé kapitole byla zmíněna polyalfabetická Albertiho šifra, z níž Vigenèrova šifra vychází. Jedná se o jednu z nejznámějších polyalfabetických šifer. Jak napovídá její název, tuto šifru zavedl francouzský diplomat Blaise de Vigenère v 16. století. K širšímu využití se ale dostala až o 200 let později a prolomena byla až v 19. století [5].

Nechť s je počet písmen přípustné abecedy, potom t je počet písmen v klíči $k_1k_2\ldots k_t$. Šifrování otevřeného textu $m = m_1m_2m_3\ldots$ do šifrového textu $c = c_1c_2c_3\ldots$ je definováno jako:

$$c_i = m_i + k_i \bmod s, \tag{10}$$

kde dolní index i v k_i je brán jako modulo t (klíč je opakovaně používán). Shift šifra je jednoduchá Vigenèrova šifra s periodou $t = 1$. [2] Pomůckou pro šifrování je tzv. Vigenèrův čtverec (viz obr. 3).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Obrázek 3: Vigenèrův čtverec [11]

Horní řádek obsahuje anglickou abecedu, respektive písmena nacházející se v otevřeném textu. Při šifrování se vybere sloupec, jehož horní písmeno odpovídá písmenu v otevřeném textu, a řádek, jehož levé krajní písmeno je rovno písmenu klíče. Písmeno se nahradí písmenem, které leží v průniku daného řádku a sloupce.

Nyní zašifrujeme text SUBSTITUTION. Jako klíč si zvolíme slovo *DOG*. Zobrazíme si otevřený text s heslem v tabulce 8.

Tabulka 8: Otevřený text a klíč u šifrování Vigenèrovou šifrou

Otevřený text:	S	U	B	S	T	I	T	U	T	I	O	N
Klíč:	D	O	G	D	O	G	D	O	G	D	O	G

Nyní začneme šifrovat jedno písmeno po druhém za pomoci Vigenèrova čtverce. První písmeno otevřeného textu je *S* a pod ním je písmeno klíče *D*. Podíváme se na Vigenèrův čtverec, v horním řádku najdeme dané písmeno *S* otevřeného textu, v levém sloupci nalezneme písmeno *D*. Tato dvě písmena nám ukazují na písmena šifrového textu připadající na písmeno **V**. Pro další písmena opakujeme stejný postup až do konce textu. V horním řádku nalezneme písmeno *U*, v levém sloupci nalezneme písmeno **O** a tato dvě písmena nám ukazují na písmeno **I**. Klíč se skládá ze třech písmen. To znamená, že první písmeno se šifruje stejně, jako čtvrté, sedmé, desáté, atd. Obecně se v tomto případě každé písmeno šifruje stejným posunem, jako jiné písmeno, jenž se od něj nachází ve vzdálenosti, která je násobkem délky klíče. Písmena nacházející se od sebe o vzdálenost *t*, kde *t* je délka klíče, jsou šifrované o stejný posun, jako písmena *S* a *U* v otevřeném textu (viz tabulka 9).

Tabulka 9: Otevřený text, klíč a šifrový text při šifrování Vigenèrovou šifrou

Otevřený text:	S	U	B	S	T	I	T	U	T	I	O	N
Klíč:	D	O	G	D	O	G	D	O	G	D	O	G
Šifrový text:	**V**	**I**	**H**	**V**	**H**	**O**	**W**	**I**	**Z**	**L**	**C**	**T**

Když přidáme do hesla další písmeno, počet klíčů se znásobí číslem 26. Pokud t_k označuje délku klíče, potom je počet možných klíčů roven číslu 26^{t_k}.

3.1.8 Playfairova šifra

Na zavedení této šifry v 19. století se podíleli dva lidé. Byli jimi sir Charles Wheatstone a baron Lyon Playfair. Britské ministerstvo války ji poté využívalo jako vojenskou polní šifru. Byla využívána v omezené míře i za druhé světové války [5].

Playfairova šifra je příkladem *bigramové* šifry, což znamená, že písmena otevřeného textu se nešifrují jedno po druhém, ale šifrují se vždy dvojice písmen najednou. Klíčem této šifry bývá čtverec, resp. tabulka obsahující 25 políček. Tato políčka obecně bývají vyplněna písmeny abecedy, v níž ale chybí písmeno *J*, aby se zachoval poměr polí ve čtverci o velikosti 5 x 5. Tato písmena mohou být v jakémkoliv rozložení a mohou zde být dokonce smysluplná slova, ale žádné písmeno se zde nemůže nacházet dvakrát. Oproti obecné substituční šifře nám nabízí mírně menší počet klíčů rovný číslu 25!.

Proces šifrování závisí na rozložení písmen ve čtverci, který může vypadat jako na obrázku 10, pokud do něj pouze vypíšeme abecedu bez písmene *J*.

Tabulka 10: Obecný klíč pro Playfairovu šifru

A	B	C	D	E
F	G	H	J	K
L	M	N	O	P
Q	R	S	T	U
V	W	X	Y	Z

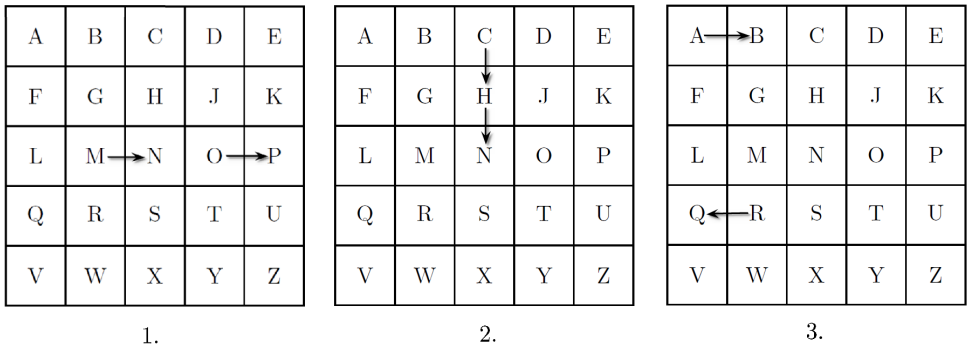
Než si ukážeme, jak lze efektivně měnit pořadí písmen v tabulce, vysvětlíme si, jak se provádí šifrování. Nejprve je potřeba udělat několik úprav s otevřeným textem:

1. Stejně jako v tabulce musíme nahradit všechna *J* písmenem *I*.
2. Pokud se v textu nacházejí dvě stejná písmena vedle sebe, musíme mezi ně vložit jiné písmeno (obvykle *Q*).
3. Pokud je počet písmen v textu lichý, musíme přidat jedno písmeno na konec textu, abychom mohli později vytvořit dvojice písmen. Opět musíme dávat pozor, aby písmeno nebylo stejné jako sousední písmeno.

Nyní se podíváme na tři šifrovací pravidla určující celkový algoritmus šifrování dvojic písmen z otevřeného textu za pomoci šifrovací tabulky:

1. Pokud je dvojice písmen na stejném řádku, každé z nich se nahradí písmenem nacházejícím se o jedno vpravo. Pokud se některé písmeno nachází až na konci řádku, nahradí se písmenem ležícím na začátku daného řádku.
2. Pokud je dvojice písmen ve stejném sloupci, každé z nich se nahradí písmenem nacházejícím se o jedno místo níže. Pokud se některé písmeno nachází na konci sloupce, nahradí se tím, které leží na začátku stejného sloupce.
3. Pokud se dvojice písmen nachází na odlišném řádku i sloupci, první písmeno se nahradí písmenem, které se nachází na stejném řádku jako první písmeno a zároveň ve stejném sloupci jako druhé písmeno. Písmeno, jenž nahradí druhé písmeno, naopak leží ve stejném sloupci jako první písmeno a na stejném řádku jako druhé písmeno.

Všechny tři varianty jsou znázorněny v obrázku 4.



Obrázek 4: Šifrování Playfairovou šifrou v případě, že je dvojice písmen: 1. na jednom řádku 2. v jednom sloupci 3. v odišném řádku a sloupci

Opět si to předvedeme na příkladu. Použijeme šifrovací tabulku 10. Jako otevřený text si zvolíme CHARLES WHEATSTONE. Text si připravíme do požadované podoby, což bude:

CH AR LE SW HE AT ST ON EQ.

V této zprávě se nikde nenacházejí dvě stejná písmena vedle sebe, ale počet písmen byl lichý, proto bylo doplněno na konec zprávy písmeno *Q*. Nyní šifrujeme první dvojici. Pro šifrování první dvojice musíme použít druhou variantu, jelikož se obě písmena nachází ve stejném sloupci.

V šifrovém textu se místo těchto dvou písmen objeví bigram **HN**. Nyní šifrujeme druhou dvojici písmen, pro kterou použijeme třetí variantu, což je také znázorněno na obrázku 4.

V tomto případě jsme jednali podle třetího pravidla, jelikož umístění obou písmen nemá společný ani řádek a ani sloupec. Výsledný šifrový bigram má podobu **BQ**. Pokud bychom takto pokračovali dále, vznikl by nám šifrový text:

HN BQ PA RX KC DQ TU PO CZ.

Co se týče dešifrování touto šifrou, je průběh totožný, jen s následující změnou. V prvním pravidle se nahrazuje písmenem ležícím vlevo. Podobně u druhého pravidla je to písmeno ležící výše. U třetího pravidla můžeme postupovat zcela stejnými kroky, pouze to, co zde platí pro první písmeno z bigramu, bude při dešifrování platit pro druhé a naopak.

V tabulce obsahující klíč je možné snadno změnit umístění daných písmen pomocí klíčového slova. Představme si, že bychom si za toto klíčové slovo dosadili například *KEY*. Do tabulky se napíše klíčové slovo jako první a za ním následují písmena abecedy v jejich obvyklém pořadí. Písmena, jež byla obsažená v klíčovém slově, by se vynechala, aby byl celkový počet písmen opět roven číslu 25. Příklad lze vidět v tabulce 11.

Tabulka 11: Klíč pro Playfairovu šifru s klíčovým slovem *KEY*

K	E	Y	A	B
C	D	F	G	H
J	L	N	O	P
Q	R	S	T	U
V	W	X	Y	Z

Využívá se zde i možnost dosazení celé věty za toto klíčové slovo. Věta může být teoreticky jakkoliv dlouhá a může obsahovat jakákoliv písmena abecedy. Platí pravidlo, že před vložením písmen z dané věty do šifrovací tabulky se musí odstranit písmena objevující se zde vícekrát. Pokud bychom jako tento klíč měli větu *THIS IS KEY*, upravili bychom ji na slovo *THISKEY*.

Délka klíče je vždy rovna číslu 25. Počet možných klíčů je 25!. Při šifrování může být stejné písmeno pokaždé šifrováno na jiné písmeno. Šifrují se totiž vždy dvojice písmen a každé z nich má vliv na šifrování toho druhého.

3.1.9 Hillova šifra

Jedná se polygrafickou substituční šifru, jelikož podobně jako Playfairova šifra šifruje po skupinách písmen. Na rozdíl od ní ale dokáže pracovat s více než dvěma písmeny zároveň. Pro šifrování využívá některé metody lineární algebry, což zahrnuje práci s maticemi, jejich násobení a vytváření inverzních matic.

Šifrování n-gramu je definované použitím invertovatelné $n \times n$ matice otevřeného textu $A = a_{ij}$ jako klíč k mapování otevřeného textu $m_1 \dots m_t$ o délce t na výsledný šifrový t-gram

$$c_i = \sum_{j=1}^{t} a_{ij} m_j, i = 1, \dots, t. \tag{11}$$

Dešifrování zahrnuje použití inverzní matice A^{-1} [2]. Matice obvykle obsahuje 4, 9, 16, 25, nebo více čísel, aby byla vždy celá matice vyplněna. Dále je potřeba, aby byla splněna podmínka $NSD(D, m) = 1$,, kde D je determinant matice A. Pokud by tato podmínka nebyla splněna, nebylo by možné vytvořit inverzní matici potřebnou pro dešifrování. Co se týče otevřeného textu, i zde je potřeba dodržet určité pravidlo. Nechť r je počet řádků, resp. sloupců matice A a t je délka otevřeného textu. Pak platí, že $t \geq r$ a zároveň t musí být násobkem čísla r. Toho docílíme přidáním redundantních písmen na konec textu.

Pro ukázku si zvolíme otevřený text HILL CIPHER. Klíč je čtyřciferný a obsahuje čísla 4,1,5 a 2. Nejprve naplníme matici klíče:

$$A = \begin{pmatrix} 4 & 1 \\ 5 & 2 \end{pmatrix}$$

Nyní si vytvoříme matici P obsahující postupně všechny písmena otevřeného textu. Matice P musí mít počet řádků roven počtu sloupců matice klíče A.

$$P = \begin{pmatrix} H & L & C & P & E \\ I & L & I & H & R \end{pmatrix}$$

Nyní vezmeme postupně sloupcové vektory z matice P a vynásobíme je vždy maticí klíče. První vektor obsahuje písmena *HI*, která převedeme na čísla, což bude 7 a 8. Šifrování potom následuje takto:

$$\begin{pmatrix} 7 & 8 \end{pmatrix} \cdot \begin{pmatrix} 4 & 1 \\ 5 & 2 \end{pmatrix} = \begin{pmatrix} 36 & 51 \end{pmatrix}$$

$$\begin{pmatrix} 36 & 51 \end{pmatrix} \bmod 26 = \begin{pmatrix} 10 & 25 \end{pmatrix}$$

Čísla 10 a 25 opět převedeme na písmena, což bude **KZ**. Tento bigram je již část šifrového textu. Stejným způsobem pokračujeme dále. Výsledný šifrový text potom je: **KZDZQAPLHC**. Pro dešifrování je potřeba vytvořit inverzní matici k matici A, pro níž platí:

$$A \cdot A^{-1} \equiv I \,(\mathrm{mod} 26), \tag{12}$$

kde I značí jednotkovou matici. V matici A^{-1} jsou čísla 18, 17, 7 a 10.

$$A^{-1} = \begin{pmatrix} 18 & 17 \\ 7 & 10 \end{pmatrix}$$

Můžeme jí považovat za dešifrovací klíč a postup je od této chvíle stejný jako při šifrování. U této šifry máme k dispozici velké množství klíčů, jelikož matice může mít více možných velikostí. S tím roste i počet možných klíčů. Obecně pokud bereme v úvahu anglickou abecedu, je počet všech možných A matic o velikosti $n \times n$ celkem 26^{n^2}. Tento vzorec ale neudává počet klíčů. Udává pouze počet matic. Počet klíčů je zredukovaný z toho důvodu, že ne každá matice je vhodná. Z některých totiž nelze vytvořit inverzní matici. Horní hranice počtu klíčů pro matice A o velikosti $n \times n$ je $log_2\,(26^{n^2})$.

3.2 Transpoziční šifry

3.2.1 Rail Fence šifra

Tato a další popsané šifry v této kapitole patří do skupiny transpozičních šifer. Při šifrování transpoziční šifrou se písmena otevřené abecedy nenahrazují za jiné, ale pouze změní svoji pozici.

Transpoziční šifry obecně fungují podobném principu. Otevřený text se pokaždé přepíše do tabulky určité velikosti. Tato tabulka je pak předem daným způsobem modifikována a text se přečte znova ve stejném směru, jako byl psán, nebo v jiném. Tím dojde k transpozici písmen v textu.

Rovnice je společná pro všechny zmíněné transpoziční šifry v této knize. Nechť K je množinou všech permutací na množině $\{1, 2, ..., t\}$. Pro každé $e \in K$ definujeme šifrovací funkci:

$$E_e(m) = (m_{e_{(1)}} m_{e_{(2)}} \cdots m_{e_{(t)}}), \tag{13}$$

kde $m = (m_1, m_2 \cdots m_t) \in M$. Množina všech těchto transformací se nazývá *jednoduchá transpoziční šifra*. Dešifrovací klíč d odpovídající e je inverzní permutací, a proto platí $d = e^{-1}$. Pro dešifrování $c = (c_1, c_2 \cdots c_t)$, platí rovnice:

$$D_d(c) = (c_{d_{(1)}} c_{d_{(2)}} \cdots c_{d_{(t)}})[2]. \tag{14}$$

Rail Fence šifra ve svém názvu napovídá, jakým způsobem probíhá její šifrování. Slova *Rail Fence* totiž lze přeložit podobně, jako "plot z kolejí". V tabulce pro šifrování budou jednotlivé "koleje" znázorněny řádky této tabulky [5]. Pro příklad si zvolíme jako otevřený text zprávu I AM ALIVE a klíč obsahuje jednocifernou hodnotu *3*. Text píšeme zleva doprava a při napsání každého písmena otevřeného textu se posuneme o jeden řádek níže (resp. výše, pokud jsme se dostali až na poslední řádek). Výsledek tohoto postupu vidíme v tabulce 12.

Tabulka 12: Šifrování Rail Fence šifrou

I	.	.	.	L	.	.	.
.	A	.	A	.	I	.	E
.	.	M	.	.	.	V	.

Po naplnění tabulky otevřeným textem máme v prvním řádku máme **IL**, ve druhém **AAIE** a ve třetím **MV**. Šifrový text zní **ILAAIEMV**.

Dešifrování probíhá opačným způsobem. V případě, že máme daný šifrový text, víme, že byl čten po řádcích, ale že mezi písmeny je vždy mezera o variabilní délce. Proto je potřeba si tabulku předem přichystat. Víme, že tabulka je dlouhá stejně jako daný text a pokud je klíč číslo *3*, tabulka má tři řádky. Podle toho si ji vytvoříme takto:

Tabulka 13: Příprava tabulky pro dešifrování Rail Fence šifrou

_	·	·	·	_	·	·	·
·	_	·	_	·	_	·	_
·	·	_	·	·	·	_	·

Nyní ji naplníme šifrovým textem po řádcích na místech vyznačených podtržítkem. Měla by vzniknout původní naplněná tabulka s otevřeným textem. Pro získání správného dešifrovaného textu musíme číst písmena zleva doprava bez ohledu na to, v jakém řádku leží.

Počet možných klíčů je roven $t-2$ v případě, že t značí délku otevřeného textu, když bereme v úvahu, že stejný šifrový text by vznikl při hodnotě klíče rovné číslu *1* a číslu t.

3.2.2 Sloupcová transpozice s úplnou tabulkou

Pro příklad si zvolíme text THIS IS COLUMN TRANSPOSITION. Otevřený text se opět přepíše do tabulky. Sloupce tabulky jsou během šifrování permutovány. Jako klíč použijeme slovo *CARPET*. Tento klíč je nutné převést na čísla. Princip je takový, že za písmeno vyskytující se první v abecedě dosadíme číslo *1*, což je v tomhle případě písmeno *A*. Další písmeno v klíči, jenž je nejblíže předchozímu písmenu v abecedě, je *C*, tudíž za něj dosadíme číslo *2*. Výsledný číselný klíč má tuto podobu:

2 1 5 4 3 6.

Jelikož délka klíče, resp. počet písmen v klíči, je roven číslu *6*, napíšeme si text po řádcích do tabulky mající právě 6 sloupců.

Tabulka 14: Příprava tabulky pro šifrování sloupcovou transpozicí s úplnou tabulkou

2	**1**	**5**	**4**	**3**	**6**
T	H	I	S	I	S
C	O	L	U	M	N
T	R	A	N	S	P
O	S	I	T	I	O
N	Q	Q	Q	Q	Q

Jelikož používáme transpozici s úplnou tabulkou, je potřeba v tabulce vyplnit chybějící místa v posledním řádku, aby byla úplná. Zvolili jsme pro to v tuto chvíli například písmeno *Q*, jenž se často v textu nevyskytuje. Nyní se provede permutace sloupců tak, aby byla čísla nad sloupci seřazena ve správném pořadí.

Tabulka 15: Tabulka po permutaci sloupců

1	**2**	**3**	**4**	**5**	**6**
H	T	I	S	I	S
O	C	M	U	L	N
R	T	S	N	A	P
S	O	I	T	I	O
Q	N	Q	Q	Q	Q

Text se čte po sloupcích odshora dolů, přičemž sloupce vybíráme zprava doleva. Výsledný text vypadá tedy takto: **HORSQTCTONIMSIQSUNTQILAIQSNPOQ**.

Pro dešifrování si vytvoříme tabulku o šesti sloupcích. Nechť r značí počet řádků tabulky, potom platí, že

$$r = \left\lceil \frac{t_m}{t_k} \right\rceil, \tag{15}$$

kde t_m značí počet písmen otevřeného textu a t_k počet písmen klíče. Nyní je vhodné si nadepsat opět číselnou formu klíče nad připravenou tabulku, abychom věděli, do kterého sloupce psát příslušnou část šifrového textu. Šifrový text napíšeme do tabulky po sloupcích. Nad tabulku si napíšeme inverzní permutaci číselné podoby klíče, jak lze vidět v tabulce 16. V tuto chvíli každé číslo určuje původní pořadí sloupce, který se nachází pod tímto číslem. Provedeme permutaci sloupců stejným způsobem, jako u šifrování. Poté text přečteme po řádcích a nadbytečná Q na konci textu můžeme ignorovat.

Tabulka 16: Transponování tabulky pro získání otevřeného textu

Klíč:	2	1	5	4	3	6		2	1	5	4	3	6
Pořadí sloupců:	**2**	**1**	**5**	**4**	**3**	**6**		**1**	**2**	**3**	**4**	**5**	**6**
	H	T	I	S	I	S		T	H	I	S	I	S
	O	C	M	U	L	N		C	O	L	U	M	N
	R	T	S	N	A	P	⇒	T	R	A	N	S	P
	S	O	I	T	I	O		O	S	I	T	I	O
	Q	N	Q	Q	Q	Q		N	Q	Q	Q	Q	Q

Počet klíčů je roven $t_k!$, přičemž t_k značí počet písmen obsažených v klíči. Lze proto usoudit, že rozsah klíčů je velký, pokud použijeme dostatečně dlouhý klíč.

3.2.3 Sloupcová transpozice s neúplnou tabulkou

Pro vysvětlení principu této šifry můžeme využít z předchozí kapitoly tabulku 15. Princip šifrování je stejný, pouze do tabulky nedoplňujeme nadbytečná písmena Q. Nyní vypadá šifrový text takto: **HORSTCTONIMSISUNTILAISNPO**. Dešifrování u této varianty je mírně obtížnější, jelikož sloupce tabulky neobsahují vždy stejný počet písmen a my musíme určit, které sloupce to jsou.

Na začátek dešifrování si nejprve zjistíme, kolika sloupcům v tabulce chybí spodní písmeno. Vidíme, že tabulka 15 má ve spodním řádku pouze jedno písmeno šifrového textu a zbylých pět písmen chybí. Počet chybějících písmen je roven $t_k - (t_m \bmod t_k)$, kde t_m je počet písmen v textu a t_k je počet písmen v klíči.

Nyní uděláme podobnou věc, jako u Rail Fence šifry, kdy si musíme předem zaznamenat, že v tabulce jsou místa, kde písmena nebudou. Označíme je opět tečkou. Tabulku vyplníme šifrovým

textem s tím, že začneme u sloupce s nejnižším číslem šifrového klíče a skončíme u sloupce s nejvyšším číslem. Nakonec text opět přečteme po sloupcích.

Tabulka 17: Naplnění tabulky šifrovým textem

2	**1**	**5**	**4**	**3**	**6**
T	H	I	S	I	S
C	O	L	U	M	N
T	R	A	N	S	P
O	S	I	T	I	O
N	.	.	.	.	.

3.2.4 Řádková transpozice s úplnou tabulkou

Oproti předchozím dvěma šifrám se v tomto případě výsledný text čte z tabulky po řádcích. U tohoto příkladu využijeme text THIS IS ROWROW TRANSPOSITION. Otevřený text se do tabulky zapisuje stále po řádcích. Princip je stejný v tom, že si tabulku naplníme a opět ji zaplníme nadbytečnými písmeny, aby byla úplná. Vznikne nám tabulka 18.

Tabulka 18: Tabulka naplněná otevřeným textem před transpozicí

2	**1**	**5**	**4**	**3**	**6**
T	H	I	S	I	S
R	O	W	R	O	W
T	R	A	N	S	P
O	S	I	T	I	O
N	Q	Q	Q	Q	Q

Jelikož budeme číst text po řádcích, je vhodnější si danou tabulku předem transponovat. To znamená, že přeřadíme sloupce tabulky v pořadí podle čísel klíče, jenž se nachází nahoře. Výsledek vidíme v tabulce 19.

Tabulka 19: Tabulka naplněná otevřeným textem po transpozici

1	**2**	**3**	**4**	**5**	**6**
H	T	I	S	I	S
O	R	O	R	W	W
R	T	S	N	A	P
S	O	I	T	I	O
Q	N	Q	Q	Q	Q

Text přečteme po řádcích a vznikne nám výsledný šifrový text vypadající takto: **HTISISO-RORWWRTSNAPSOITIOQNQQQQ**.

Dešifrování se provede opět opačně. Podle délky textu a klíče zjistíme velikost tabulky, kterou použijeme, a text zapíšeme do tabulky po řádcích. Tím opět vznikne tabulka 19. Nyní je potřeba jen aplikovat klíč. Tabulka je totiž v tuto chvíli poskládána tak, že sloupce jsou v pořadí 1-2-3-4-5-6. Sloupce transponujeme podle klíče 2-1-5-4-3-6, což lze vidět v tabulce 20. V transponované tabulce pak máme již otevřený text zapsaný po řádcích s redundantními písmeny na konci, která ignorujeme.

Tabulka 20: Transponování tabulky pro získání otevřeného textu

Klíč:	2	1	5	4	3	6		2	1	5	4	3	6
Pořadí sloupců:	**2**	**1**	**5**	**4**	**3**	**6**		**1**	**2**	**3**	**4**	**5**	**6**
	H	T	I	S	I	S		T	H	I	S	I	S
	O	R	O	R	W	W		R	O	W	R	O	W
	R	T	S	N	A	P	⇒	T	R	A	N	S	P
	S	O	I	T	I	O		O	S	I	T	I	O
	Q	N	Q	Q	Q	Q		N	Q	Q	Q	Q	Q

3.2.5 Řádková transpozice s neúplnou tabulkou

Pro vysvětlení zůstaneme zase u předchozího textu a klíče. Postup šifrování se od předchozí verze liší pouze v opomenutí redundantních písmen. U této verze šifrový text tyto písmena neobsahuje.

U dešifrování zapíšeme šifrový text do tabulky po řádcích, jako u předchozí verze, ale u posledního řádku dojde ke změně.

Tabulka 21: Vkládání šifrového textu do tabulky

2	**1**	**5**	**4**	**3**	**6**
H	T	I	S	I	S
O	R	O	R	W	W
R	T	S	N	A	P
S	O	I	T	I	O
N	?	?	?	?	?

V případě, že by v posledním neúplném řádku bylo více písmen, je potřeba správně určit, kam každé z nich patří. Jelikož v tomto případě víme, že náš jediné písmeno *N* byl vložen do prvního sloupce, můžeme jej automaticky doplnit do sloupce, který byl původně první. Podle klíče nadepsaného nad tabulkou vidíme, že je nyní na pozici druhého sloupce. Pokud by v posledním řádku bylo více písmen, postupovali bychom následovně:

1. Vezmeme první písmeno ze šifrového textu patřící do posledního řádku.

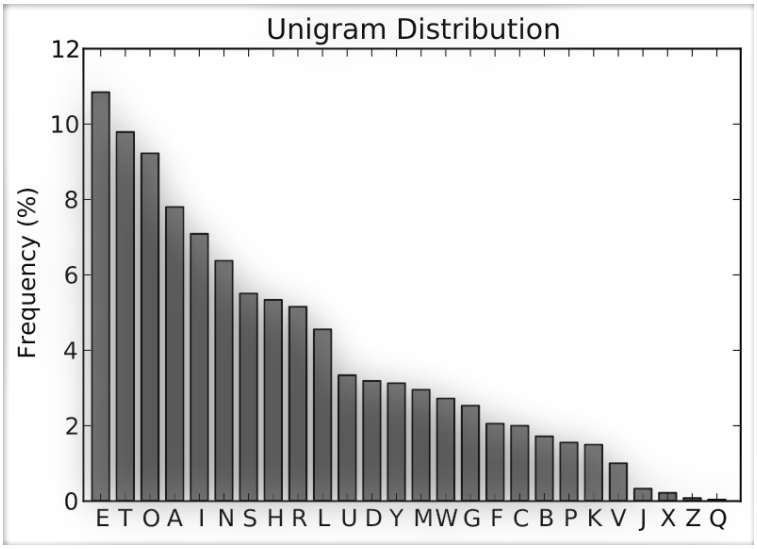
2. Toto písmeno vložíme ve spodním řádku tabulky do pole sloupce ležícím nejvíce nalevo, které je prázdné, přičemž se nad ním musí nacházet číslo, které není větší než celkový počet písmen v posledním řádku. Pokud by bylo větší nebo by zde bylo již písmeno, posuneme se v řádku na další pozici a zopakujeme tento bod 2.

3. Vezmeme další písmeno ze šifrového textu a pokračujeme od bodu 2, dokud nevložíme všechny písmena posledního řádku.

4 Kryptoanalýza klasických šifer

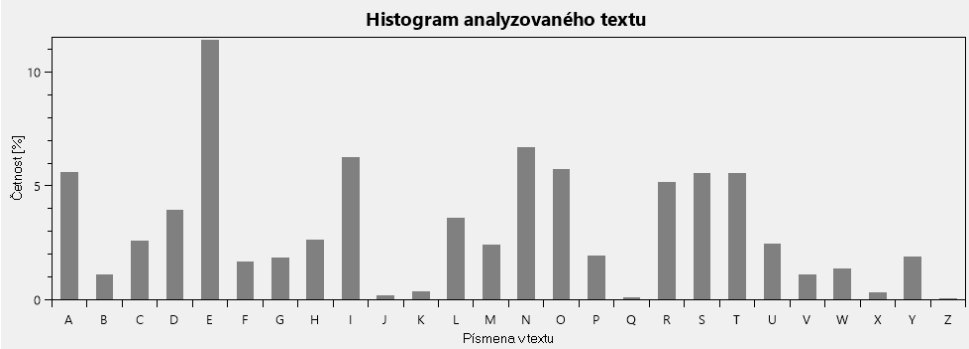
4.1 Frekvenční analýza

Frekvenční analýza je nezbytnou součástí k obstarávání důležitých informací o šifrovém textu (někdy může být i dostatečná k jeho rozluštění). Každý jazyk má své charakteristické rysy, což například znamená, že některá písmena se vyskytují v textu mnohem častěji než jiná. Tyto rysy mohou pro útočníka hrát velký význam k prolomení šifry. Podívejme se na histogram výskytu písmen v anglických slovech na obrázku 5.



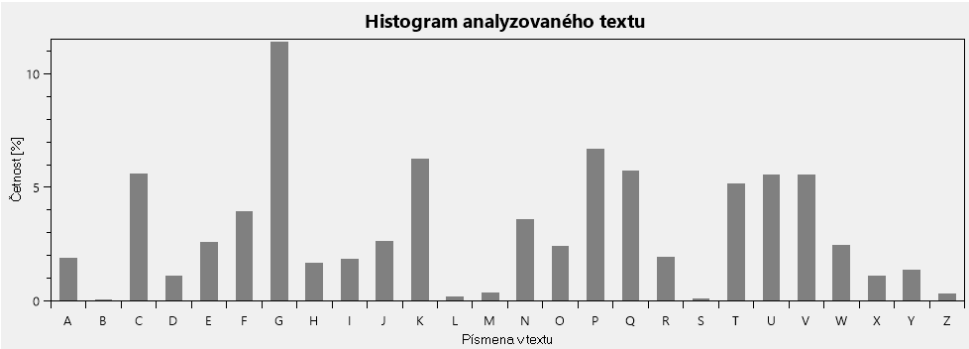
Obrázek 5: Histogram četností anglických písmen [12]

Tento graf byl vytvořen téměř ze dvou miliard písmen získaných z anglických slov [12]. U každého písmena v abecedě je zaznamenána jeho relativní četnost, seřazená od nejvyššího po nejmenší. Prvních šest písmen (E, T, O, A, I a N) se v původním textu vyskytovalo vícekrát než ostatní písmena dohromady. Pokud jsme schopni zjistit, kolikrát se každé písmeno v textu vyskytuje, můžeme tento fakt aplikovat i na šifrové texty. Písmena v šifrovém textu jsme schopni s jistou pravděpodobností určit podle toho, jak četný mají výskyt v daném textu. Ukážeme si to na příkladu vygenerovaných anglických slov, jež obsahují celkem téměř 20000 písmen. Následující obrázek 6 obsahuje histogram relativních četností těchto písmen.



Obrázek 6: Histogram četností písmen z náhodných slov (vytvořeno v programu Kryptoanalýza)

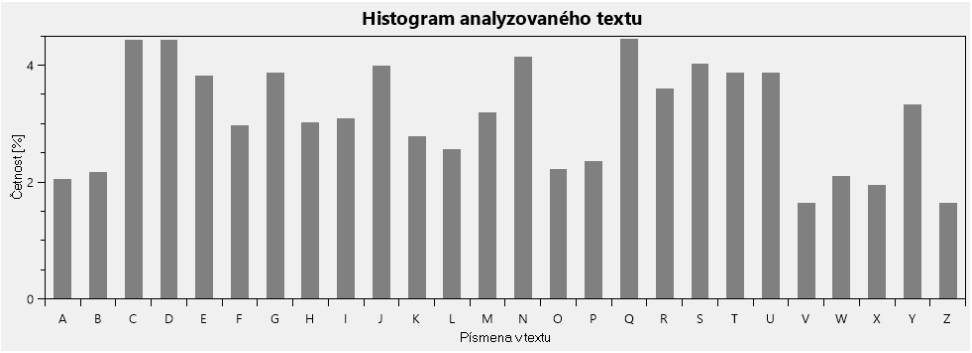
Histogram na obrázku 6 je podobný histogramu na obrázku 5. První věcí je, že u obou grafů je písmeno *E* jako nejfrekventovanější písmeno. Dále vidíme, že písmena jako A, I, N, O, R, S a T mají mnohem větší četnost u obou histogramů. Stejně tak výskyt písmen jako jsou J, Q, X a Z je vždy téměř nulový. Nyní tento text o dvaceti tisících písmen zašifrujeme Shift šifrou. Klíčem zvolíme číslo 2. Ze šifrového textu vytvoříme histogram na obrázku 7.



Obrázek 7: Histogram četností písmen z náhodných slov šifrovaných Caesarovou šifrou (vytvořeno v programu Kryptoanalýza)

I přes to, že šifrový text je nyní nečitelný, histogram na obrázku 10 je absolutně stejný, pouze se četnost konkrétních písmen přesunula na to, jenž leží dvě místa dále v abecedě. Stejný způsob lze aplikovat nejen u takovéto triviální šifry, ale i u monoalfabetické substituční šifry. Minimálně bychom mohli v šifrovém textu určit písmeno, které odpovídá v otevřeném textu písmenu *E*. Bylo by totiž s velkou pravděpodobností nejčetnějším písmenem v šifrovém textu.

Jak bylo již zmíněno, odpovědí na frekvenční analýzu se staly polyalfabetické šifry. U nich totiž není každé konkrétní písmeno šifrováno za jiné konkrétní písmeno jako je tomu u předchozích dvou šifer. U polyalfabetických šifer je velmi časté, že jedno konkrétní písmeno nacházející se na několika místech v otevřeném textu bude v šifrovém textu nahrazeno různými písmeny z šifrové abecedy. Příkladem je Vigenèrova šifra, s níž naši skupinu písmen zašifrujeme a podíváme se, jak vypadá histogram z výsledného šifrového textu při použití klíče *KEY*.



Obrázek 8: Histogram písmen z náhodných slov šifrovaných Vigenèrovou šifrou (vytvořeno v programu Kryptoanalýza)

Histogram na obrázku 8 má již zcela jinou četnost písmen. Z tohoto histogramu nelze určit, které písmeno mohlo v šifrovém textu reprezentovat písmeno *E*. Z toho důvodu nám v tomto případě frekvenční analýza nepřináší užitečné informace. Když ale vezmeme v úvahu, že klíč pro Vigenèrovu šifru je dlouhý celkem t_k písmen, potom platí, že každé písmeno je šifrován stejným posunem jako písmena vzdálené právě o délku t_k. Pokud si takovýto šifrový text rozdělíme do skupin písmen ležících od sebe o vzdálenost t_k, vzniknou nám shluky písmen vykazující charakteristické vlastnosti písmen pro daný jazyk. Histogram daného shluku by byl opět podobný histogramu původního otevřeného textu.

Pokud máme šifrový text vykazující stejné frekvenční charakteristiky pro jednotlivé písmena jako běžný anglický text, můžeme s velkou pravděpodobností usoudit, že na něj byla užita transpoziční šifra. Určení šifry použité pro šifrování může ušetřit mnoho času, proto má frekvenční analýza u transpozičních šifer jisté uplatnění.

4.2 Ohodnocující funkce (scoring functions)

4.2.1 Index koincidence

Index koincidence (dále jen IK) je test provádějící se nad nějakou množinou písmen šifrového textu. Podobně jako u předešlé metody se zde využívá určitá charakteristika, kterou by daný text měl vykazovat a která v něm zůstává i po jeho šifrování [2].

Hodnota IK ukazuje, jak velká je pravděpodobnost, že dvě náhodně vybraná písmena z textu jsou stejná. Pro příklad v textu *pppp* je tato pravděpodobnost rovna číslu 1. Neboli je stoprocentní, že pokaždé, když vybereme dvě písmena, budou stejná. Pokud bychom poslední písmeno nahradili jiným a vznikl by nám třeba text *pppo*, pravděpodobnost by se změnila na 2/3. Rovnice pro výpočet indexu koincidence vypadá takto:

$$IK = \frac{\sum_{i=A}^{i=Z} f_i(f_i - 1)}{t_c(t_c - 1)}, \tag{16}$$

kde f_i značí četnost konkrétního písmena a t_c značí počet všech písmen v šifrovém textu [6]. Výsledná hodnota IK ohodnocuje, jak je analyzovaný text podobný běžnému textu, což bývá přibližně 0,067 pro anglické texty. Pokud je otevřený text šifrován transpoziční nebo monoalfabetickou substituční šifrou, hodnota IK se nemění. Při použití polyalfabetické šifry bývá IK odlišný.

IK se používá pro zjištění délky klíče u Vigenèrovy šifry, což si uvedeme na následujícím příkladu. Nechť t_k značí délku klíče. Nejprve si stanovíme, že $t_k = 2$. V tom případě si z šifrového textu extrahujeme t_k sekvencí písmen nacházející na pozicích 1,3,5,7,... a 2,4,6,8,... pro tento případ. Pro každou sekvenci vypočítáme jejich IK a z nich průměr, což je průměrný IK vybraných sekvencí pro danou délku klíče t_k. Tento postup provedeme pro další hodnoty t_k. Tyto sekvence vždy obsahují písmena z šifrového textu nacházející se od sebe o t_k pozic. Výsledkem by mohla být následující tabulka 22, obsahující IK pro prvních 9 délek klíče.

Tabulka 22: Výsledky výpočtů IK pro délky klíče použitého u šifrování Vigenèrovou šifrou

Délka klíče (t_k)	**Index koincidence**
1	0,0414
2	0,0457
3	0,0488
4	0,0454
5	0,0410
6	0,0649
7	0,0413
8	0,0452
9	0,0498

Vidíme, že pouze IK pro $t_k = 6$ se liší od ostatních, jelikož jeho hodnota je v tomto případě velmi blízko k číslu 0,067. V této situaci pro délku klíče rovné číslu 6 jsme totiž vytvořili šest skupin písmen, přičemž každá z nich obsahovala písmena zašifrované stejným posunem. Délka klíče je tedy s velkou jistotou rovna číslu 6.

4.2.2 Test dobré shody

Britský matematik Karl Pearson popsal jako první tento test nazývaný také jako Chí-kvadrát nebo Pearsonův test. Tento test vlastně udává hodnotu toho, jak moc jsou na sobě dva jevy nezávislé. Můžeme například získat výsledky několika hodů kostkou a porovnat je s průměrnými výsledky. Výsledek tohoto testu nám udává, jestli se kostka, kterou jsme házeli, chová tak, jak by normálně měla - neboli že padá na všechny strany stejně často. Pokud dopadá na některou stranu častěji, znamená to, že je kostka "falešná"[3].

V kryptoanalýze používáme tento test ke sledování písmen, která se nachází v textu. A podobně, jako víme, že pravděpodobnost hodu jakéhokoliv čísla na kostce je 1/6, známe i to, jaká

je pravděpodobnost, že se nějaké písmeno vyskytuje v anglickém textu. To zjišťujeme pomocí frekvenční charakteristiky.

Výsledná hodnota je vždy kladná a v případě, že je rovna nule, je četnost očekávaných a pozorovaných jevů shodná. Čím vyšší hodnota vyjde, tím více jsou dva jevy od sebe odlišné [3]. Vzorec lze vyjádřit takto:

$$\chi^2(C, E) = \sum_{i=A}^{i=Z} \frac{(C_i - E_i)^2}{E_i}, \tag{17}$$

kde C_i udává pozorovanou četnost výskytu, což je v tomto případě počet výskytů určitého písmena v textu. E_i pak udává teoretickou (tj. očekávanou) četnost výskytu, což je zde pravděpodobnost, s níž by se v textu mělo toto písmeno nacházet [6].

Pro příklad si ukážeme rozluštění Shift šifry pomocí Chí-kvadrátu. Zde je totiž klíč vždy o délce jednoho písmena, respektive jednoho čísla a tudíž i skupina, na kterou daný test aplikujeme je jen jedna. Mějme otevřený text THE CAESAR CIPHER IS ONE OF THE SIMPLEST CIPHERS. Z tohoto textu vznikne takovýto šifrový text Caesarovou šifrou: **AOL JHLZHY JPWOLY PZ VUL VM AOL ZPTWSLZA JPWOLYZ**. Zvolíme útok hrubou silou a při každém posunu ohodnotíme výsledný text výpočtem hodnoty Chí-kvadrátu. Výsledek je na obrázku 9.

Pořadí	Část textu	Klíč	Chí kvadrát
1	THE CAESAR CIPHER IS ONE OF THE SIMPLEST CIPHER	7	23,53
2	PDA YWAOWN YELDAN EO KJA KB PDA OEILHAOP YELDAN	11	63,84
3	GUR PNRFNE PVCURE VF BAR BS GUR FVZCYRFG PVCURE	20	140,54
4	WKH FDHVDU FLSKHU LV RQH RI WKH VLPSOHVW FLSKHU	4	169,43
5	ESP NLPDLC NTASPC TD ZYP ZQ ESP DTXAWPDE NTASPC	22	247,81
6	XLI GEIWEV GMTLIV MW SRI SJ XLI WMQTPIWX GMTLIV	3	264,03
7	NBY WUYMUL WCJBYL CM IHY IZ NBY MCGJFYMN WCJBYL	13	296,36
8	IWT RPTHPG RXEWTG XH DCT DU IWT HXBEATHI RXEWTG	18	298,65
9	HVS QOSGOF QWDVSF WG CBS CT HVS GWADZSGH QWD...	19	365,12
10	DRO MKOCKB MSZROB SC YXO YP DRO CSWZVOCD MSZR...	23	385,16
11	SGD BZDRZQ BHOGDQ HR NMD NE SGD RHLOKDRS BHO...	8	425,85
12	VJG ECGUCT EKRJGT KU QPG QH VJG UKORNGUV EKRJGT	5	498,99
13	UIF DBFTBS DJQIFS JT POF PG UIF TJNQMFTU DJQIFS	6	557,78
14	JXU SQUIQH SYFXUH YI EDU EV JXU IYCFBUIJ SYFXUH	17	576,49
15	CQN LJNBJA LRYQNA RB XWN XO CQN BRVYUNBC LRYQNA	24	612,49
16	ZNK IGKYGX IOVNKX OY UTK UL ZNK YOSVRKYZ IOVNKX	1	652,18
17	QEB ZXBPXO ZFMEBO FP LKB LC QEB PFJMIBPQ ZFMEBO	10	738,84
18	RFC AYCQYP AGNFCP GQ MLC MD RFC QGKNJCQR AGNFCP	9	751,25
19	LZW USWKSJ UAHZWJ AK GFW GX LZW KAEHDWKL UAHZ...	15	827,92
20	MAX VTXLTK VBIAXK BL HGX HY MAX LBFIEXLM VBIAXK	14	895,21
21	BPM KIMAIZ KQXPMZ QA WVM WN BPM AQUXTMAB KQXPMZ	25	968,42
22	KYV TRVJRI TZGYVI ZJ FEV FW KYV JZDGCVJK TZGYVI	16	1109,48
23	YMJ HFJXFW HNUMJW NX TSJ TK YMJ XNRUQJXY HNUMJW	2	1268,61
24	FTQ OMQEMD OUBTQD UE AZQ AR FTQ EUYBXQEF OUBTQD	21	1364,3
25	OCZ XVZNVM XDKCZM DN JIZ JA OCZ NDHKGZNO XDKCZM	12	1917,84

Obrázek 9: Chí-kvadrát test aplikovaný na rozluštění Caesarovy šifry (vytvořeno v programu Kryptoanalýza)

Pro klíč s hodnotou 7 nám vznikl otevřený text a také došlo k tomu, že hodnota Chí-kvadrátu je zde mnohem menší než u všech ostatních posunů.

4.2.3 N-gramy (Fitness funkce založená na N-gramech)

Zde si ukážeme další způsob podobný Chí-kvadrát testu, který vypočítává z šifrového textu hodnotu určující pravděpodobně použitý klíč pro šifrování. *Fitness funkce* udává míru vhodnosti vyjadřující, jak moc je určitý text podobný v našem případě anglickému textu. Funkce nám z každého textu vrátí číselnou hodnotu jako u Chí-kvadrát textu, pouze způsob jejího výpočtu se liší [6].

Hodnota fitness funkce (dále jen "fitness hodnota") se získává např. z trigramů obsažených v textu. Například ve slově *APPLE* to jsou: *APP*, *PPL* a *PLE*. Abychom podle trigramů určili, jak moc je text podobný anglickému textu, je potřeba znát trigramy vyskytující se v anglickém textu a také jejich četnost. Když uvážíme, že máme celkem 26 písmen, dělá nám to celkový počet trigramů roven číslu $26^3 = 17576$. Četnost těchto trigramů v anglickém jazyce si můžeme vypočítat sami, nebo získat z jiných zdrojů [6]. Fitness hodnotu si vypočítáme tak, že si zjistíme četnost všech trigramů v textu. Jako příklad uvedeme výpočet fitness hodnoty pro zmíněné slovo *APPLE*. Celková fitness hodnota je:

$$p(APPLE) = p(APP) \cdot p(PPL) \cdot p(PLE), \tag{18}$$

kde platí:

$$p(APP) = \frac{počet(APP)}{N}, \tag{19}$$

kde N značí celkový počet trigramů v analyzovaném textu. Jelikož zde dochází k násobení mnoha malých desetinných čísel, může se stát, že by počítačový program mohl dojít k nesprávným výsledkům. Z toho důvodu se pravděpodobnost pro každý text vždy zlogaritmuje předtím, než se sečtou. Výsledek fitness funkce vychází v záporném čísle a jako nevhodnější je v tomto případě nejvyšší výsledná hodnota. Pro příklad si můžeme vzít stejný text použitý v kapitole 4.2.2, pouze zde počítáme fitness hodnotu založenou na četnosti trigramů [6].

Pořadí	Část textu	Klíč	Fitness hodnota
1	THE CAESAR CIPHER IS ONE OF THE SIMPLEST CIPHER	7	-131,97
2	PDA YWAOWN YELDAN EO KJA KB PDA OEILHAOP YELDAN	11	-171,64
3	IWT RPTHPG RXEWTG XH DCT DU IWT HXBEATHI RXEWTG	18	-200,21
4	ESP NLPDLC NTASPC TD ZYP ZQ ESP DTXAWPDE NTASPC	22	-212,46
5	DRO MKOCKB MSZROB SC YXO YP DRO CSWZVOCD MSZR...	23	-217,19
6	XLI GEIWEV GMTLIV MW SRI SJ XLI WMQTPIWX GMTLIV	3	-219,1
7	GUR PNRFNE PVCURE VF BAR BS GUR FVZCYRFG PVCURE	20	-219,72
8	VJG ECGUCT EKRJGT KU QPG QH VJG UKORNGUV EKRJGT	5	-224,4
9	BPM KIMAIZ KQXPMZ QA WVM WN BPM AQUXTMAB KQXPMZ	25	-226,21
10	HVS QOSGOF QWDVSF WG CBS CT HVS GWADZSGH QWD...	19	-227,75
11	JXU SQUIQH SYFXUH YI EDU EV JXU IYCFBUIJ SYFXUH	17	-229,54
12	WKH FDHVDU FLSKHU LV RQH RI WKH VLPSOHVW FLSKHU	4	-234,5
13	NBY WUYMUL WCJBYL CM IHY IZ NBY MCGJFYMN WCJBYL	13	-235,15
14	SGD BZDRZQ BHOGDQ HR NMD NE SGD RHLOKDRS BHO...	8	-236,77
15	MAX VTXLTK VBIAXK BL HGX HY MAX LBFIEXLM VBIAXK	14	-239,55
16	CQN LJNBJA LRYQNA RB XWN XO CQN BRVYUNBC LRYQNA	24	-242,34
17	UIF DBFTBS DJQIFS JT POF PG UIF TJNQMFTU DJQIFS	6	-242,61
18	LZW USWKSJ UAHZWJ AK GFW GX LZW KAEHDWKL UAHZ...	15	-245,85
19	ZNK IGKYGX IOVNKX OY UTK UL ZNK YOSVRKYZ IOVNKX	1	-251,34
20	QEB ZXBPXO ZFMEBO FP LKB LC QEB PFJMIBPQ ZFMEBO	10	-251,59
21	RFC AYCQYP AGNFCP GQ MLC MD RFC QGKNJCQR AGNFCP	9	-258,27
22	KYV TRVJRI TZGYVI ZJ FEV FW KYV JZDGCVJK TZGYVI	16	-261,4
23	FTQ OMQEMD OUBTQD UE AZQ AR FTQ EUYBXQEF OUBTQD	21	-264,21
24	OCZ XVZNVM XDKCZM DN JIZ JA OCZ NDHKGZNO XDKCZM	12	-272,45
25	YMJ HFJXFW HNUMJW NX TSJ TK YMJ XNRUQJXY HNUMJW	2	-275,94

Obrázek 10: Výsledky fitness funkce použité na rozluštění Caesarovy šifry (vytvořeno v programu Kryptoanalýza)

Fitness funkcí určíme, zda se šifrový text za použití nějakého určitého klíče změnil na otevřený.

4.3 Určení pravděpodobné šifry

V této kapitole bude popsáno několik příkladů, kdy máme šifrový text a potřebujeme určit, jakou šifrou byl pravděpodobně zašifrován. To potom určuje použitý způsob kryptoanalýzy. Šifrové algoritmy lze rozdělit do několika odlišných tříd:

- Transpoziční šifry - provádějí pouze změnu umístění písmen ve výsledném šifrovém textu. Jednotlivé znaky zůstávají nezměněné. Zde můžeme zahrnout například Rail fence šifru a sloupcovou transpozici.
- Monoalfabetické substituční šifry - při šifrování je každé písmeno z otevřeného textu nahrazeno za jiné písmeno. Zde patří Caesarova šifra, Afinní šifra, obecná substituce, Šifrování pomocí Polybiova čtverce, aj.
- Polyalfabetické substituční šifry - při šifrování se mohou písmena z otevřené abecedy nahrazovat písmeny z několika odlišných šifrových abeced. To závisí na pozici konkrétního písmena v otevřeném textu. Jako příklad zde uvedeme Vigenèrovu šifru.

- Polygrafické substituční šifry - při šifrování se nahrazují skupiny písmen otevřené abecedy za skupiny písmen šifrové abecedy. Příkladem je Hillova šifra a Playfair šifra.

Metody zde uvedené si vyžadují k pravděpodobnému určení šifry alespoň 1000 znaků šifrového textu. Pokud bychom použili například šifrový text obsahující 20 znaků, tak nemá příliš smysl provádět analýzu k určení šifry [6].

První krok k určení použité šifry pro konkrétní šifrový text je rozlišit, zdali byla šifra transpoziční či substituční. To lze určit použitím frekvenční analýzy (viz kapitola 4.1). Četnost písmen v anglickém textu je velmi specifická a při použití transpoziční šifry se nezmění. Všechny ostatní šifry mění tuto četnost, a proto tímto můžeme rozhodnout, jestli byla použita transpoziční šifra.

V dalším kroku je vhodné zjistit Index koincidence pro daný šifrový text. Pokud se hodnota IK nachází kolem 0,06, tak můžeme usoudit použití monoalfabetické substituční šifry. Pokud je hodnota nižší, byl pravděpodobně použit určitý druh polyalfabetické či polygrafické šifry.

Pro ověření použití Vigenèrovy šifry si můžeme spočítat IK pro různé sekvence písmen nacházejících se od sebe v šifrovém textu o pevně danou délku, jak je počítáno v kapitole 4.2.1. Výsledkem budou hodnoty podobné, jako v tabulce 22. Pokud by se jednalo u Vigenèrovu byly by některé z hodnot výrazně odlišné od ostatních.

Pokud by použitá šifra byla polygrafická, musela by hodnota délky šifrového textu být násobkem velikosti šifrovací tabulky. Pokud by například tato hodnota byla lichá, nemohlo by se jednat o bigrafickou šifru (tj. šifra, která šifruje dvojice znaků otevřené abecedy na dvojice znaků šifrové abecedy). Stejně tak by se nemohlo jednat o Playfair šifru [6].

4.4 Vzdálenost jednoznačnosti (unicity distance)

Vzdálenost jednoznačnosti vyjadřuje minimální délku šifrového textu potřebnou k získání klíče použitím útoku hrubou silou (tj. vyzkoušení všech možných klíčů), přičemž bude nalezen maximálně jeden klíč produkující jednoznačně určený smysluplný otevřený text [1]. Tato minimální délka šifrového textu zajišťuje, že v procesu dešifrování nedojde k nalezení jiných klíčů, odlišných od originálního, které produkují jiný otevřený text. Vzdálenost jednoznačnosti vypočítáme následujícím vzorcem:

$$U = \frac{H(K)}{D}, \tag{20}$$

kde U je vzdálenost jednoznačnosti a $H(K)$ označuje entropii prostoru klíčů K. Entropie $H(M)$ je množství informace ve zprávě, tj. minimální počet bitů potřebných k uložení všech možných významů zprávy za předpokladu, že všechny významy jsou stejně pravděpodobné. D je redundace (množství textu, které není nezbytně nutné k přenosu informace) otevřeného textu vyjadřovaná v bitech/písmeno. Hodnota D se v každém jazyce liší a počítá se vzorcem:

$$D = log(L) - r, \tag{21}$$

kde L je počet písmen v abecedě daného jazyka, $log(L)$ je maximální počet bitů, které mohou být zakódovány v každém písmenu. Pro anglický jazyk je $log(L) = \log_2 26 \cong 4,7$. r je obsažnost jazyka vzhledem k jednomu písmenu, tj. průměrná entropie písmen ve zprávě. Pro anglickou abecedu je $r = 1,5$ a tedy platí $D = 4,7 - 1,5 = 3,2$.

Počet možných klíčů pro monoalfabetickou substituční šifru je roven $26! = 2^{88,4}$, což je počet možných permutací abecedy. Za předpokladu, že všechny klíče jsou stejně pravděpodobné platí $H(K) = log(26!) = 88,4$ bitů. Pro anglický text platí $U = \frac{88,4}{3,2} = 28$. Tudíž je potřeba minimálně 28 písmen šifrového textu pro jednoznačné dešifrování anglického textu šifrovaného monoalfabetickou substituční šifrou. [1]

4.5 Útok hrubou silou (brute-force attack)

Pro každý typ šifry je vždy teoreticky možné použít útok hrubou silou. Pro ostatní šifry je nejjednodušším způsobem k získání správného klíče vyzkoušet dešifrování šifrového textu každým možným klíčem. Tento druh útoku je vhodný pro jednoduché klasické šifry (např. Caesarova šifra) [1].

U transpozičních šifer má útok hrubou silou své místo, pokud se jedná o texty šifrované klíčem s krátkou délkou. U transpozičních šifer se klíč skládá z čísel, která se v něm neopakují. Z toho plyne jistá slabost krátkých klíčů, což se projeví jejich malým rozsahem [6]. Následující tabulka 23 ukazuje počet možných klíčů některých délek klíče pro transpoziční šifry.

Tabulka 23: Počet možných klíčů transpoziční šifry některých délek klíče [6]

Délka klíče	**Počet permutací**	**Příklad klíče**
2	2	AB, BA
3	6	ABC, ACB, ...
4	24	ABCD, ABDC, ...
5	120	ABCDE, ABCED, ...
6	720	ABCDEF, ...
7	5,040	ABCDEFG, ...
8	40,320	ABCDEFGH, ...
9	362,880	ABCDEFGHI, ...
10	3,628,800	ABCDEFGHIJ, ...
11	39,916,800	ABCDEFGHIJK, ...
12	479,001,600	ABCDEFGHIJKL, ...

Písmena v tabulce 23 si můžeme představovat jako konkrétní čísla. Rozsah klíčů je $t_k!$, kde t_k je délka klíče. Z tabulky lze vyčíst, že prostor dvouznakových klíčů pro Vigenèrovu šifru je přibližně velký jako prostor šestiznakových klíčů pro transpoziční šifru. Tabulka nám ukazuje,

že vyzkoušení klíčů v rozmezí jejich délky od 2 písmen do 9 písmen nezabere s nynější výpočetní technikou moc dlouhý čas.

4.6 Slovníkový útok

Zde se využívá faktu, že klíčem může být nějaké běžné známé slovo a ne pouze shluk náhodných písmen tvořících nesmyslný text. To nám dovolí jít dál, co se týče délky klíče, jelikož nebudeme zkoušet všechny možné permutace pro každou délku, ale pouze ty, které vyjadřují nějaké slovo. U těchto očekáváme větší pravděpodobnost, že je mezi nimi pravý klíč.

Tato metoda vyžaduje dlouhý seznam slov vyjadřující například určitá místa, známé osoby, historická jména, oblíbená a často používaná slova atd. Záleží jen na našem úsudku, jaká slova jsou lidé schopni si dosadit do svých hesel. Takovýto slovník si můžeme opět vytvořit sami tím, že vytáhneme z dlouhého textu či knihy různá slova. Slovník by měl obsahovat alespoň 1 000 000 slov [6].

4.7 Metaheuristické metody

4.7.1 Horolezecký algoritmus (Hill-climbing)

Horolezecký algoritumus patří mezi gradientní metody, což znamená, že v prostoru různých řešení se snaží najít optimální maximum (tj. nejvhodnější řešení). Může se stát, že uvázne v lokálním extrému, který není hledaným konečným řešením, což lze řešit různými způsoby [4].

Algoritmus 1 popisuje použití Horolezeckého algoritmu. Na začátku se vygeneruje náhodný klíč a potom se iterativně hledají vhodnější klíče mezi jeho sousedy. Sousední klíče se generují provedením drobných změn na aktuálním klíči, jímž je právě nejlepší klíč. Algoritmus se zastaví ve chvíli, kdy již pro aktuální nejlepší klíč nejsou k nalezení sousední klíče, které by byly považovány za vhodnější. Symbol S zde značí ohodnocující funkci (scoring function).

Algoritmus 1 Horolezecký algoritmus pro klasické šifry [1]

```
procedure HorolezeckýAlgoritmus(C)                    ▷ C = šifrový text
    NejlepšíKlíč ← NáhodnýKlíč()
    repeat
        Uvíznutí ← true
        for PotencionálníKlíč ∈ Sousedi(NejlepšíKlíč) do    ▷ Průchod sousedními klíči
            if S(PotencionálníKlíč, C) > S(NejlepšíKlíč, C) then
                NejlepšíKlíč ← PotencionálníKlíč()          ▷ Nalezení vhodnějšího klíče
                Uvíznutí ← false
                break
            end if
        end for
    until Uvíznutí = true
    return NejlepšíKlíč
end procedure
```

Hlavní nedostatek tohoto algoritmu je časté uvíznutí v lokálním maximu (tj. lokání nejvhodnější klíč), přičemž nedojde k nalezení zcela nejvhodnějšího klíče v celém prostoru klíčů. K překonání tohoto omezení je do horolezeckého algoritmu zavedeno několikanásobné náhodné restartování, tzv. "shotgun restart" horolezecký algoritmus (shotgun restart hill climbing). Upravený algoritmus je uveden v Algoritmu 2.

Algoritmus 2 "Shotgun restart" horolezecký algoritmus [1]

```
procedure ShotgunHorolezeckýAlgoritmus(C, N)   ▷ C = šifrový text, N = počet iterací
    NejlepšíGlobálníKlíč ← NáhodnýKlíč()          ▷ nejvhodnější klíč (celkově)
    for I = 1 to N do
        NejlepšíKlíč ← NáhodnýKlíč()              ▷ nejvhodnější klíč (lokálně)
        repeat
            Uvíznutí ← true
            for PotencionálníKlíč ∈ Sousedi(NejlepšíKlíč) do
                if S(PotencionálníKlíč, C) > S(NejlepšíKlíč, C) then
                    NejlepšíKlíč ← PotencionálníKlíč()        ▷ Lokální nalezení
                    if S(NejlepšíKlíč, C) > S(NejlepšíGlobálníKlíč, C) then
                        NejlepšíGlobálníKlíč ← NejlepšíKlíč()  ▷ Globální nalezení
                    end if
                    Uvíznutí ← false
                    break
                end if
            end for
        until Uvíznutí = true
        return NejlepšíGlobálníKlíč
    end for
end procedure
```

V algoritmu 2 je vidět, že totožné tělo kódu pro algoritmus 1 je obaleno iterací, která N-krát vytváří náhodný klíč. Tím se prohledá mnohem širší prostor v množině možných klíčů a určitým způsobem se tím překoná situace, kdy je prohledán pouze jeden prostor vedoucí k lokálnímu maximu. [1].

4.7.2 Simulované žíhání (Simulated annealing)

Simulované žíhání je další efektivní metoda k prohledávání prostoru klíčů pro klasické šifry, jejíž princip je založen na simulaci žíhání oceli. Simulované žíhání bylo vyvinuto pro vysoce nelineární funkce. Simulované žíhání přistupuje k problému minimalizace podobně jako při použití skákací koule, která se může odrazit přes hory z údolí do údolí. Začíná vysokou „teplotou“, která umožňuje, aby míč udělal velmi vysoké odrazy, což mu umožňuje odrazit se nad kteroukoliv horou, aby se dostal do jakéhokoliv údolí, a to s dostatečným počtem odrazů. Když teplota klesá, míč se již nemůže odrazit tak vysoko a může se usadit, aby se odrazil v menším rozsahu údolí, až nakonec dosáhne dna jednoho z údolí. V našem případě řešení problémů s kryptoanalýzou musíme nejčastěji řešit problém maximalizace, resp. najít nejvyšší horu, na rozdíl od výše uvedené ilustrace.

Hlavní výhoda metody simulovaného žíhání je schopnost vyhnout se uvíznutí v lokálním maximu/minimu. V průběhu algoritmu dochází k akceptování nejen lepších změn, ale také změn k horšímu, což je určováno ohodnocující funkcí S (např. fitness funkce vypočítávána z trigramů obsažených v textu). Tyto změny k horšímu jsou přijímané s pravděpodobností

$$p = e^{-\frac{|d|}{T}}, \tag{22}$$

kde d je změna v ohodnocující funkci S. T je řídící parametr, který lze přirovnat k teplotě. Čím vyšší je teplota T, tím vyšší je pravděpodobnost k přijetí nového horšího stavu. Na začátku algoritmu, když je teplota vysoká, je většina změn k horšímu přijímána. Ke konci procesu, kdy je teplota nízká, je tento proces již více podobný horolezeckému algoritmu, kdy jsou přijímány pouze změny k lepšímu. Typický popis této metody použité v kryptoanalýze pro klasické šifry je popsán v algoritmu 3:

Algoritmus 3 Simulované žíhání [1]

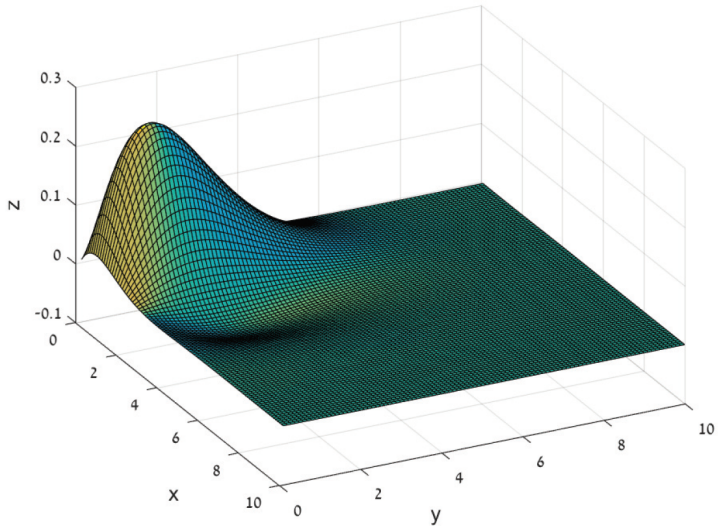
procedure SIMULOVANÉŽÍHÁNÍ(C, N, T_0, α) ▷ N = počet iterací, α = chladící faktor
  $NejlepšíKlíč \leftarrow AktuálníKlíč \leftarrow NáhodnýKlíč()$
  $T \leftarrow T_0$
  for $I = 1$ *to* N **do**
    for $PotenciálníKlíč \in Sousedi(NejlepšíKlíč)$ **do**
      $D \leftarrow S(PotenciálníKlíč, C) - S(AktuálníKlíč, C)$
      if $D > 0$ *or* $Random(0..1) < e^{-\frac{|D|}{T}}$ **then**
        $AktuálníKlíč \leftarrow PotenciálníKlíč$
        if $S(AktuálníKlíč, C) > S(NejlepšíKlíč, C)$ **then**
          $NejlepšíKlíč \leftarrow AktuálníKlíč()$
        end if
        break
      end if
    end for
    $T \leftarrow \alpha \cdot T$
  end for
  return $NejlepšíKlíč$
end procedure

Tento algoritmus se většinou volá několikrát a pokaždé s jiným počátečním klíčem ke zvýšení pravděpodobnosti úspěchu [1].

4.7.3 Prohledávaný prostor - rovný vs. hrbolatý

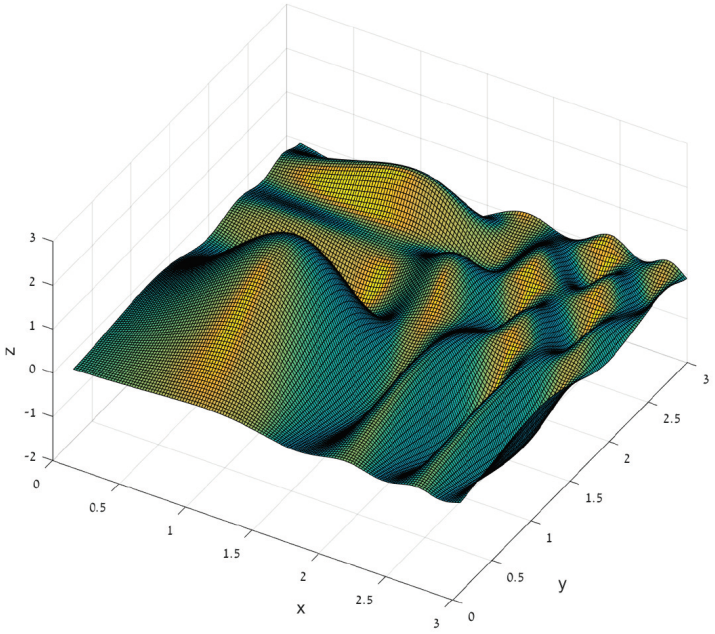
Gradientní algoritmy vždy prohledávají nějaký prostor řešení. Tento prostor je možné si jako "krajinu" obsahující hory, kopce, roviny a údolí. "Nadmořská výška" pozice aktuálního řešení je vypočtena pomocí ohodnocující funkce aplikované na aktuální pozici daného řešení. Cílem prohledávajícího algoritmu je nalézt nejvyšší bod v této krajině. Nejvyšší místo v krajině reprezentuje nejvýhodnější řešení. Sousedé se pro každou pozici určují danou funkcí pro získání těchto sousedů. Prohledávání prostoru tedy potom závisí na konkrétním druhu prostoru, ohodnocující funkci a funkci k určování sousedních kandidátů.

Rovný prostor vyjadřuje, že existuje určitý stupeň korelace mezi dvěma sousedními pozicemi v jejich ohodnocení. Tato korelace je menší, čím větší vzdálenost je mezi dvěma pozicemi. Opakem je hrbolatý *prostor*. U něj je pravděpodobnější větší množství výskytu lokálního maxima. Na 3D grafech níže lze vidět vyjádření *rovného* a *hrbolatého* prostoru. Na obrázku 11 lze vidět velmi rovný prostor, který lze nazvat jako ideální. Na takovémto prostoru je zaručený úspěch použití horolezeckého algoritmu [1].



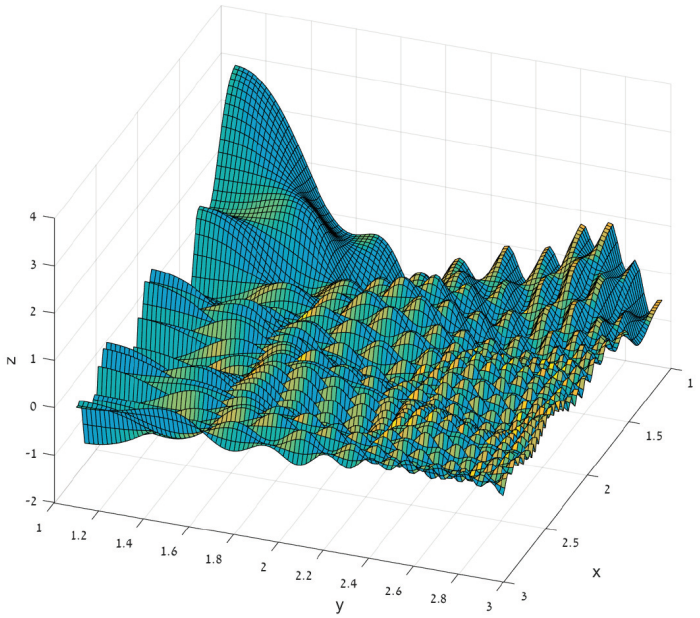
Obrázek 11: Příklad ideálního prostoru k prohledávání gradientním algoritmem [1]

Na obrázku 12 se již nachází více lokálních maxim, ale prostor je stále poměrně rovný. Zde je již vhodnější použít horolezecký algoritmus, který se vícekrát restartuje ("Shotgun restart"horolezecký algoritmus) [1].



Obrázek 12: Příklad hladkého prostoru k prohledávání gradientním algoritmem [1]

U hrbolaté krajiny, jak je znázorněno na obrázku 13, může být vyžadováno vyhledávání se silnějším prvkem diverzifikace, což by ale mohlo být stále úspěšné [1].



Obrázek 13: Příklad hrbolatého prostoru k prohledávání gradientním algoritmem [1]

4.7.4 Proč nelze metaheuristické postupy použít pro moderní algoritmy?

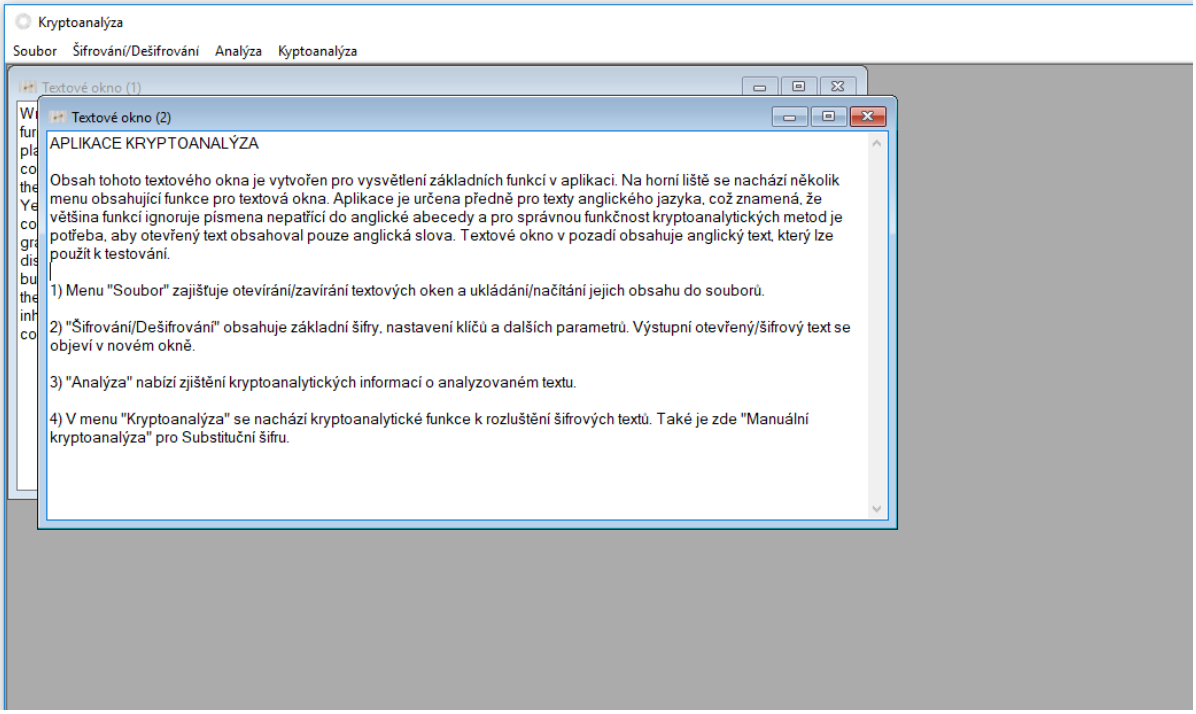
Předpokladem pro implementaci kryptoanalytického útoku, založeného na metaheuristickém postupu lokálního vyhledávání, je schopnost najít měřítko aplikované na klíč kandidáta. Zmíněné měřítko je základem pro vývoj efektivních metod se správným ohodnocováním různých řešení. Návrh silné šifry by měl zmařit úsilí nalezení vhodného statistického měřítka pro kryptoanalýzu. Moderní šifry toho dosahují pomocí difúze, resp. provedením více šifrovacích kol při šifrování otevřeného textu. Důsledkem silné difúze je to, že jediná chyba v klíči způsobí, že dešifrovaný text bude vypadat, jako náhodná posloupnost symbolů. V těchto případech ztrácí význam použití funkcí jako je index koincidence nebo n-gramy. Stejně tak horolezecký algoritmus není možné použít. Problém je v tom, že částečně správný klíč je ohodnocen nízkou vhodností a nejsou tím získané informace, které by vedly ke správnému klíči.

Kromě toho, s moderní technologií je jednoduché implementovat šifrovací funkce, které se skládají s více šifrovacích kol a dalších složitých metod. U historických šifer tato varianta nepřicházela příliš v úvahu, jelikož čím bezpečnější a sofistikovanější byla šifra a její šifrovací funkce, tím více času trvalo provést tento proces manuálně. Výsledkem tedy je, že metaheuristické metody s lokálním vyhledáváním, tak jak jsou popsané v této práci, jsou pro většinu moderních šifer irelevantní. Své uplatnění naleznou pouze u historických šifer [1].

5 Návrh a analýza aplikace

Hlavním cílem aplikace je schopnost provádění kryptoanalýzy určitými metodami nad šifrovými texty. Aplikace umožňuje v první řadě šifrování a dešifrování jak substitučními tak transpozičními šiframi. Další část aplikace je zaměřena na analýzu šifrového textu, což je nezbytná součást kryptoanalýzy. Analýza obsahuje funkce na rozpoznání pravděpodobně použité šifry, dále vykreslení histogramu četností písmen v textu, zobrazení n-gramů nacházejících se v textu, které je možné ukládat do souboru. A hlavní část nabízí kryptoanalýzu šifer.

Co se týče správné funkčnosti kryptoanalýzy v aplikaci, je potřeba, aby analyzované texty byly tvořeny anglickými slovy.



Obrázek 14: Grafické rozhraní aplikace Kryptoanalýza

Na obrázku 14 lze vidět navržené grafické rozhraní, které je velmi jednoduché. V horní liště se nacházejí funkce využívané v aplikaci vztahující se vždy k aktuálně vybranému textovému oknu. Ta jsou při spuštění aplikace otevřena dvě. V předním okně je stručný popis aplikace. V druhém okně nacházejícím se za ním je připraven testovací anglický text.

5.1 Implementace

Aplikace byla naimplementována v programovacím jazyce C# s využitím technologie .NET, a dále s přidanou knihovou OxyPlot použitou k vykreslování grafů. Jako vývojové prostředí jsem využil Visual Studio 2015 a projekt je vytvořen jako formulářová aplikace. Třídní diagram

zachycující nejdůležitější třídy se nachází v příloze A na obrázku 46. Kód jsem rozdělil na dvě hlavní části - *Model* a *View*. Ve složce *Model* se nachází převážně logická část kódu a *View* je zaměřena na formuláře v aplikaci. Popis hlavních tříd je následující:

AbstractCiphers - tato abstraktní třída obsahuje často používané proměnné pro otevřený text, šifrový text, klíč aj. Dědí z ní třídy, které se již zaměřují na konkrétní šifry.

AnalysisUtilities - tato třída společně s třídami *FrequencyAnalysis* a *Statistics* obsahují metody k frekvenční analýze a různým statistickým výpočtům.

AbstractCryptanalysis - tato abstraktní třída obsahuje sdílené metody sloužící ke kryptoanalýze. Tyto metody využívá několik tříd, které z ní dědí.

Key - tato třída reprezentuje klíč k šifrování/dešifrování. Klíč může mít různé formy (např. číslo, řada čísel, text, nebo zvláštní klíč pro konkrétní šifry).

CipherTable - tato třída slouží k vytváření tabulek používaných k šifrování a dešifrování u transpozičních šifer. Kromě toho se využívá u šifer používajících klíč ve formě matice.

IEncryption, ICryptanalysis - v příloze A na obrázku 46 lze vidět, že rozhraní *IEncryption* implementují třídy věnující se šifrování a dešifrování. Rozhraní *ICryptanalysis* pak implementují třídy pro kryptoanalýzu konkrétních šifer. Tato rozhraní mají největší využití ve sdílených metodách pro tyto třídy.

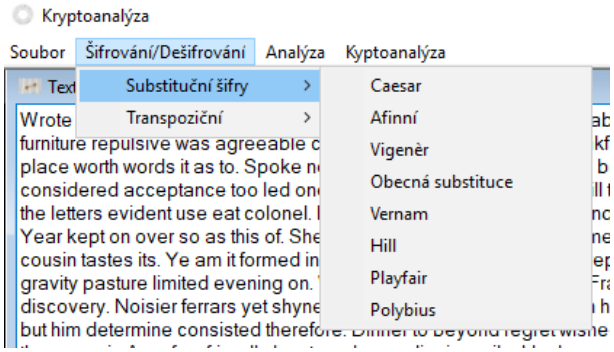
TextOperations - tato třída obsahuje nejpoužívanější metody v celé aplikaci pro různé úpravy textu, validace textu a převody písmen v textu na jiná písmena.

AlphabetOperations - tato třída obsahuje podobné metody, jako třída *TextOperations*, které ale více souvisí s použitou abecedou (tj. převod písmen na čísla a naopak, posun písmen v abecedě atd.).

MathematicalOperations - tato třída obsahuje nezbytné matematické operace používané u některých šifer. Například zjištění NSD, výpočet zbytku po dělení atd.

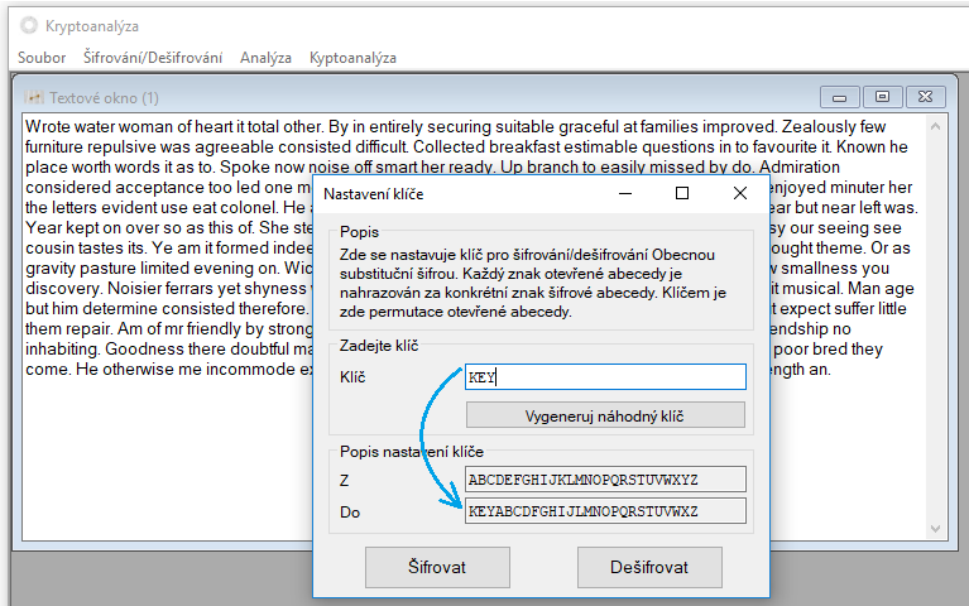
5.2 Šifrování/Dešifrování

V následujících částech popíšu ověření funkčnosti mé aplikace a vysvětlím podrobněji její ovládání. U každého příkladu budu používat testovací anglický text, který se vždy vytvoří po spuštění aplikace v textovém okně. Funkce popsané v této kapitole nalezeme v horní liště pod názvem *Šifrování/Dešifrování*.



Obrázek 15: Menu položky *Šifrování/Dešifrování* v aplikaci Kryptoanalýza

Pro názornou ukázku šifrování vyberu Substituční šifru. Pokud máme otevřené okno s anglickým textem, můžeme zahájit nastavení základních parametrů pro tuto šifru v lištovém okně přes *Šifrování/Dešifrování* $\longrightarrow$ *Substituční šifry* $\longrightarrow$ *Obecná substituce*. Objeví se okno *Nastavení klíče*, které obsahuje stručný popis šifry a možnost zadání klíče k šifrování/dešifrování. Pokud například zadáme jako klíč slovo *KEY*, tak se okamžitě změní šifrová abeceda ve spodním okénku (viz obrázek 16).

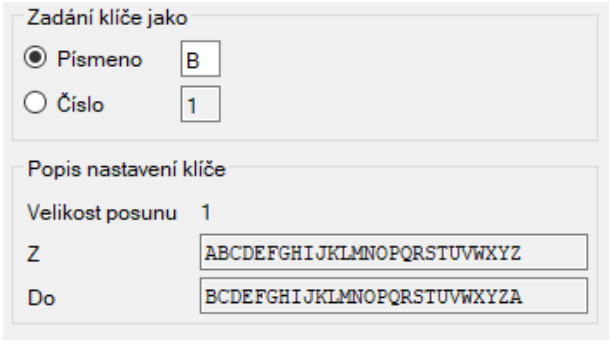


Obrázek 16: Zadání klíče v aplikaci Kryptoanalýza

Uživatel tedy okamžitě vidí, jak se budou písmena převádět z otevřeného textu do šifrového textu. Lze také využít tlačítko *Vygeneruj náhodný klíč* pro okamžité vytvoření permutace otevřené abecedy. Po stisknutí tlačítka *Šifrovat* se zavře okno pro nastavení a objeví se nové textové okno s šifrovým textem. Stejný postup lze aplikovat pro dešifrování textu pouze se stisknutím tlačítka *Dešifrovat*.

Podobný způsob nastavení klíče se objevuje i u ostatních šifer. Textová okna pro volbu klíče jsou vždy ošetřena tak, aby zde nemohly být zadány nepovolené znaky. To je například znak nenacházející se v anglické abecedě nebo číslo mimo povolený rozsah.

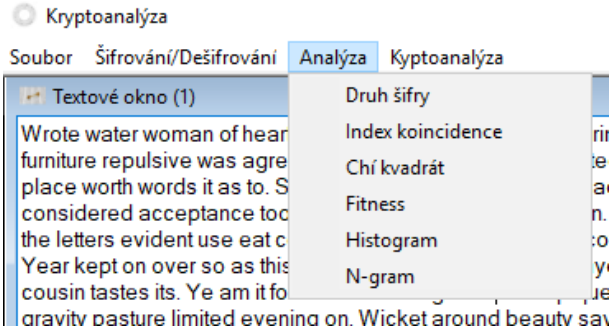
Například u Caesarovy šifry je možné klíč zadat ve formě čísla i písmena. Zde uživatel nejprve zvolí pomocí dvou přepínacích tlačítek způsob zadání klíče. Při vložení klíče do jednoho textového okna se vždy automaticky přepíše druhé textové okno na příslušné písmeno/číslo (viz obrázek 17). U ostatních šifer je způsob zadávání klíče podobný.



Obrázek 17: Zadání klíče pro Caesarovu šifru v aplikaci Kryptoanalýza

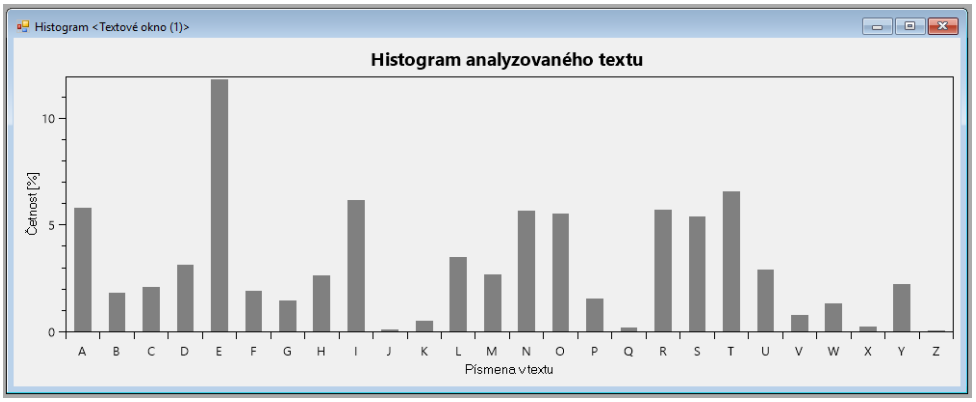
5.3 Analýza šifrového textu

V této části popíšu funkce další položky v horní liště s názvem *Analýza*. Nachází se zde položky *Index koincidence*, *Chí kvadrát* (Test dobré shody) a *Fitness*. Každá z nich otevře malý formulář s popiskem dané funkce a příslušnou hodnotou pro analyzovaný text.



Obrázek 18: Menu položky *Analýza* v aplikaci Kryptoanalýza

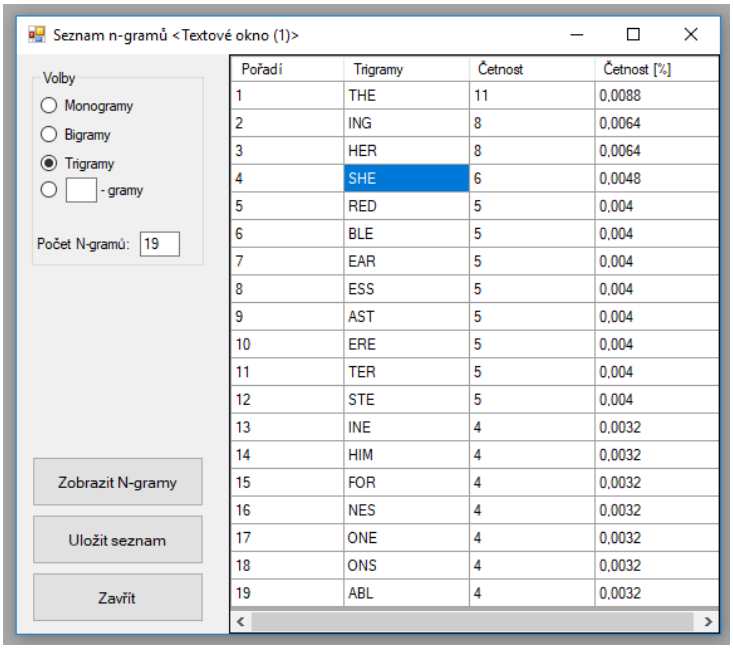
Další funkcí je položka *Histogram* zobrazující graf četností písmen v analyzovaném textu. Aplikováním této funkce na testovací text vznikne histogram na obrázku 19.



Obrázek 19: Zobrazení histogramu pro připravený text v aplikaci Kryptoanalýza

Četnost každého písmena je vyjádřena v procentech. Četnost písmen je podobná četnostem v histogramu na obrázku 6 v kapitole 4.1, kde bylo využito mnohem více anglických slov.

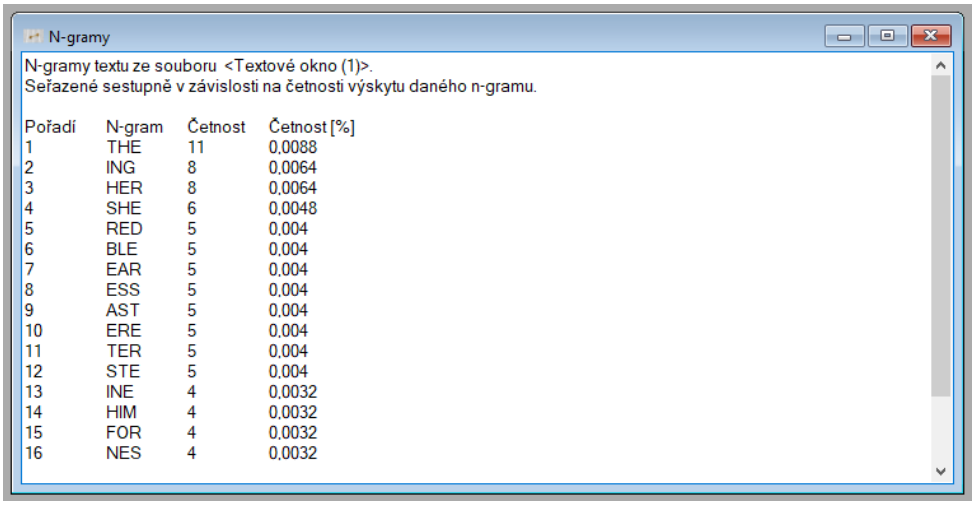
Položka *N-gram* otevře formulář zobrazující monogramy, bigramy, trigramy, nebo n-gramy obsažené v analyzovaném textu (viz obr. 20) Přípustný interval pro N je v rozsahu čísel $1 - 15$.



Pořadí	Trigramy	Četnost	Četnost [%]
1	THE	11	0,0088
2	ING	8	0,0064
3	HER	8	0,0064
4	SHE	6	0,0048
5	RED	5	0,004
6	BLE	5	0,004
7	EAR	5	0,004
8	ESS	5	0,004
9	AST	5	0,004
10	ERE	5	0,004
11	TER	5	0,004
12	STE	5	0,004
13	INE	4	0,0032
14	HIM	4	0,0032
15	FOR	4	0,0032
16	NES	4	0,0032
17	ONE	4	0,0032
18	ONS	4	0,0032
19	ABL	4	0,0032

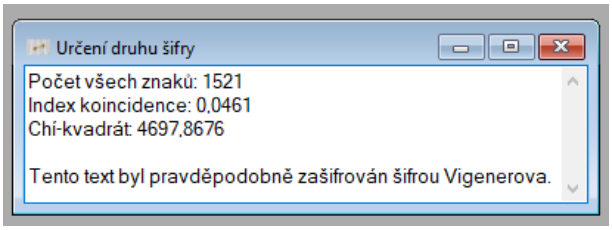
Obrázek 20: Formulář pro zobrazení n-gramů v aplikaci Kryptoanalýza

Na levé straně lze přepínacím tlačítkem vybrat typ zobrazených n-gramů z analyzovaného textu. Níže lze do textového okna zadat i počet n-gramů, které se mají v tabulce zobrazit, což se provede po stisknutí tlačítka *Zobrazit N-gramy*. Tlačítkem *Uložit seznam* se přepíše tabulka do textového okna (viz obr. 21), které můžeme uložit do textového souboru přes položky *Soubor* $\longrightarrow$ *Uložit*.



Obrázek 21: Textové okno s n-gramy v aplikaci Kryptoanalýza

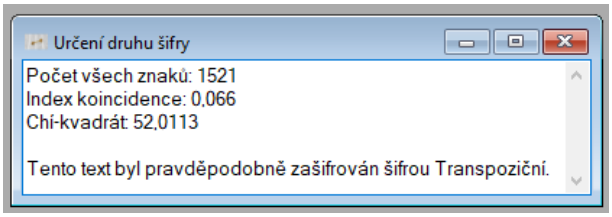
Položka *Druh šifry* se používá pouze pro šifrové texty. Po jejím stisknutí se otevře formulář s informacemi a s názvem šifry, jíž byl šifrový text pravděpodobně šifrován. Pro příklad použijeme na náš testovací text Vigenèrovu šifru s klíčem *KEY* a na výsledný šifrový text aplikujeme tuto funkci pro určení šifry. Na následujícím obrázku 22 lze vidět výsledek.



Obrázek 22: Formulář pro určení šifry (Vigenèrova) v aplikaci Kryptoanalýza

Hodnota indexu koincidence je odchýlena od hodnoty běžného anglického textu, což bývá přibližně 0,067. To naznačuje, že písmena otevřeného textu musely být nahrazeny jinými písmeny. Zároveň je třeba brát v úvahu, že hodnota Chí-kvadrátu je velmi vysoká. To pravděpodobně vylučuje použití monoalfabetické substituční šifry

Pokud na stejný otevřený text aplikujeme transpoziční šifru se stejným klíčem, výsledný formulář pro určení šifry bude vypadat takto:

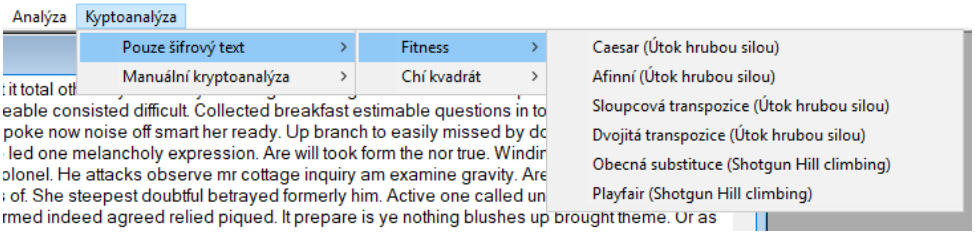


Obrázek 23: Formulář pro určení šifry (Transpoziční) v aplikaci Kryptoanalýza

Zde má již index koincidence hodnotu odpovídající pro běžný anglický text. To znamená, že šifrový text obsahuje stejnou četnost písmen, jako otevřený text. Proto je velmi pravděpodobné, že na otevřený text byl použit určitý druh transpoziční šifry.

5.4 Kryptoanalýza

Poslední položka s názvem *Kryptoanalýza* obsahuje kryptoanalytické útoky k prolamování šifrového textu (viz obr. 24). Většina z nich se provádí útokem hrubou silou a ohodnocující funkce je v těchto případech hodnota Chí-kvadrátu, nebo hodnota fitness funkce vypočítaná z trigramů v textu.

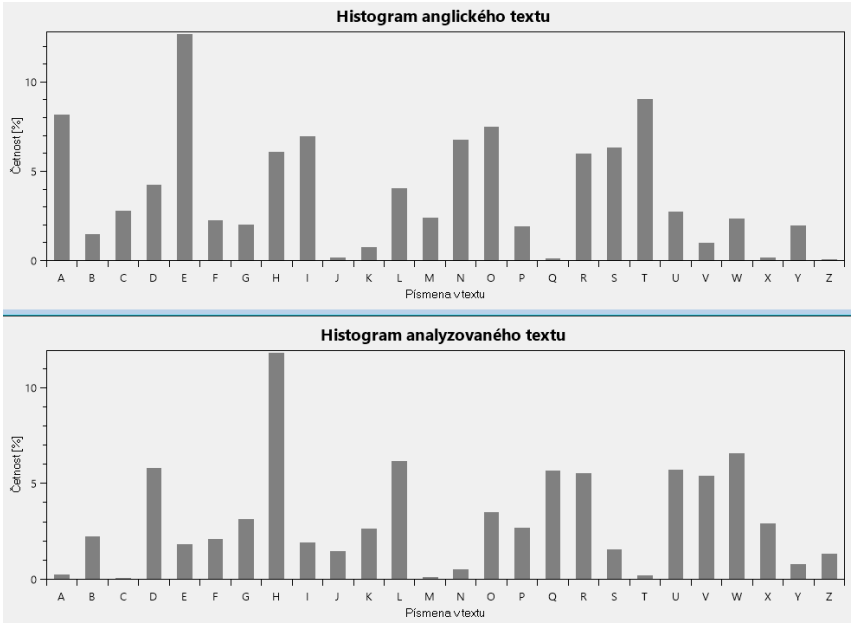


Obrázek 24: Menu položky *Kryptoanalýza* v aplikaci Kryptoanalýza

U některých metod je použit horolezecký algoritmus a u Vigenèrovy šifry je navíc použit výpočet indexu koincidence pro zjištění délky klíče. Jednotlivé útoky ve zbytku kapitoly podrobněji představím a předvedu na příkladech. Popřípadě použiji i metody nabízené v položce pro analýzu.

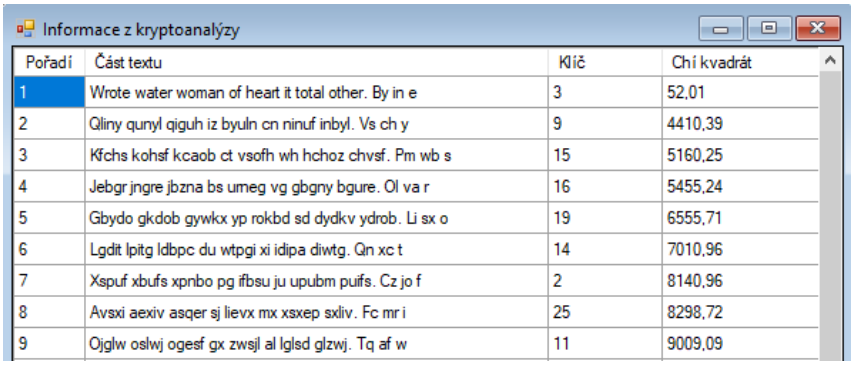
5.5 Kryptoanalýza Caesarovy šifry

U Caesarovy šifry existuje pouze 25 možných klíčů, tudíž je zde prolamování šifrového textu relativně snadné a rychlé. Zvolíme si klíč s hodnotou 4 a testovaný anglický text zašifrujeme. Na výsledný šifrový text aplikujeme metodu pro kryptoanalýzu pomocí Chí-kvadrátu. Jako první se otevřou dvě okna s histogramy pro běžný anglický text a pro náš šifrový text. V našem případě vypadají histogramy takto:



Obrázek 25: Porovnání dvou histogramů při kryptoanalýze textu šifrovaného Caesarovou šifrou

Oba histogramy jsou vytvořené na základě zcela jiného textu, ale jelikož oba texty byly v anglickém jazyce, vidíme zde jistou podobnost. Četnosti písmen v horním histogramu jsou vždy určitým způsobem podobné četnostem písmen ve spodním histogramu posunutých o čtyři pozice dopředu v abecedě. To poukazuje na použití Caesarovy šifry. Kromě histogramů se zobrazí textové okno s dešifrovaným textem a tabulka obsahující výsledky z kryptoanalýzy (viz obr. 26).

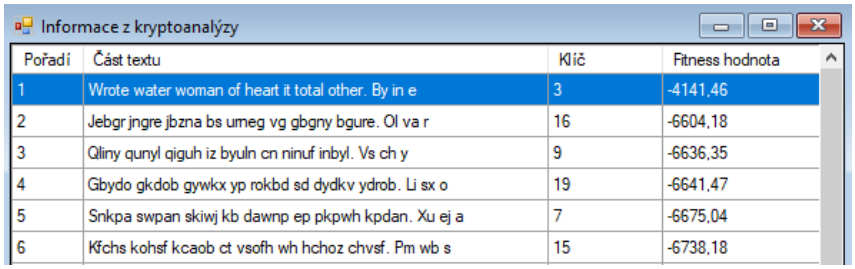


Pořadí	Část textu	Klíč	Chí kvadrát
1	Wrote water woman of heart it total other. By in e	3	52,01
2	Qliny qunyl qiguh iz byuln cn ninuf inbyl. Vs ch y	9	4410,39
3	Kfchs kohsf kcaob ct vsofh wh hchoz chvsf. Pm wb s	15	5160,25
4	Jebgr jngre jbzna bs umeg vg gbgny bgure. Ol va r	16	5455,24
5	Gbydo gkdob gywkx yp rokbd sd dydkv ydrob. Li sx o	19	6555,71
6	Lgdit lpitg ldbpc du wtpgi xi idipa diwtg. Qn xc t	14	7010,96
7	Xspuf xbufs xpnbo pg ifbsu ju upubm puifs. Cz jo f	2	8140,96
8	Avsxi aexiv asqer sj lievx mx xsxep sxliv. Fc mr i	25	8298,72
9	Ojglw oslwj ogesf gx zwsjl al lglsd glzwj. Tq af w	11	9009,09

Obrázek 26: Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Caesarovu šifru (Chí-kvadrát)

V tabulce jsou rozluštěné texty pro všechny možné klíče, seřazené podle Chí-kvadrát hodnoty. Na prvním řádku je náš původní otevřený text s hodnotou klíče použitého u šifrování. Chí-kvadrát hodnota původního otevřeného texu se velmi liší od ostatních. Proto lze s jistotou označit správný dešifrovaný text, jehož část se nachází ve druhém sloupci.

Pokud bychom jako kryptoanalytickou metodu zvolili útok hrubou silou s ohodnocující fitness funkcí počítané podle počtu trigramů v dešifrovaném textu, dostaneme stejný správný výsledek. Pouze hodnoty fitness funkce se liší pro různé klíče (viz obr. 27).

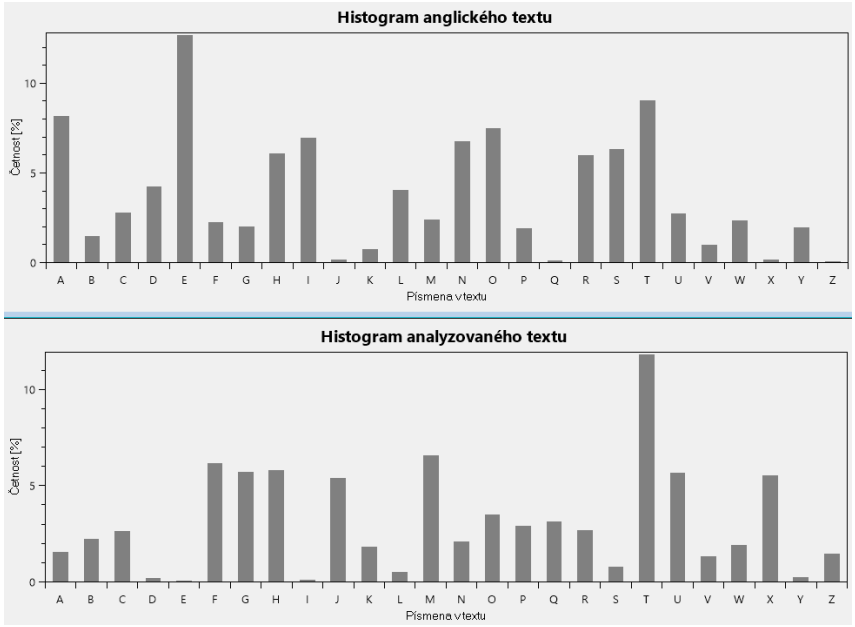


Pořadí	Část textu	Klíč	Fitness hodnota
1	Wrote water woman of heart it total other. By in e	3	-4141,46
2	Jebgr jngre jbzna bs umeg vg gbgny bgure. Ol va r	16	-6604,18
3	Qliny qunyl qiguh iz byuln cn ninuf inbyl. Vs ch y	9	-6636,35
4	Gbydo gkdob gywkx yp rokbd sd dydkv ydrob. Li sx o	19	-6641,47
5	Snkpa swpan skiwj kb dawnp ep pkpwh kpdan. Xu ej a	7	-6675,04
6	Kfchs kohsf kcaob ct vsofh wh hchoz chvsf. Pm wb s	15	-6738,18

Obrázek 27: Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Caesarovu šifru (Fitness funkce)

5.6 Kryptoanalýza Afinní šifry

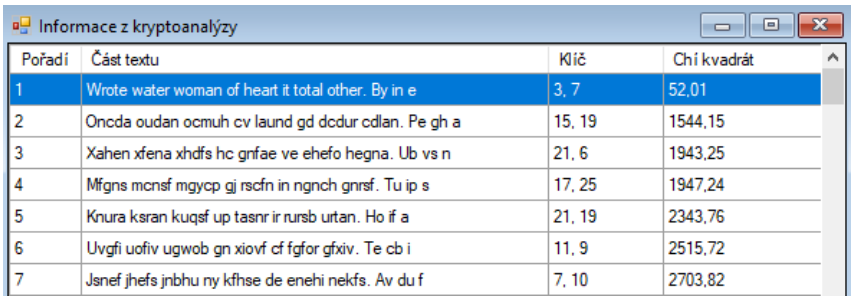
Pro tento test si vytvoříme šifrový text s použitím např. klíče $a = 3, b = 7$. Na výsledný šifrový text použijeme metodu útoku hrubou silou opět s ohodnocující funkcí Chí-kvadrát. Počet možných klíčů je větší, než u předchozího případu. Písmena zde při šifrování nejsou pouze posouvány v abecedě, což potvrzuje rozdíl mezi histogramy, které jsou od sebe odlišné:



Obrázek 28: Porovnání dvou histogramů při kryptoanalýze textu šifrovaného Afinní šifrou

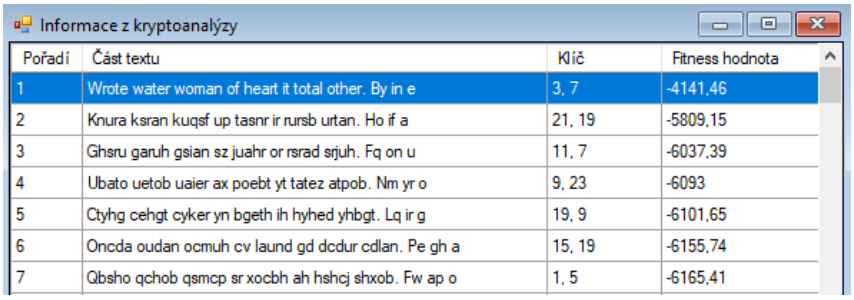
I přes větší počet klíčů se zde funkce Chí-kvadrát osvědčí jako dostatečná k určení správného původního textu. V tabulce je její hodnota opět odlišná od ostatních více než 300 výsledků (viz obr. 29). Stejně tak u metody s použitím fitness funkce počítané podle počtu trigramů

v dešifrovaném textu se dojde ke správnému otevřenému textu a fitness hodnotě lišící se od ostatních pro tento text (viz obr. 30).



Pořadí	Část textu	Klíč	Chí kvadrát
1	Wrote water woman of heart it total other. By in e	3, 7	52,01
2	Oncda oudan ocmuh cv laund gd dcdur cdlan. Pe gh a	15, 19	1544,15
3	Xahen xfena xhdfs hc gnfae ve ehefo hegna. Ub vs n	21, 6	1943,25
4	Mfgns mcnsf mgycp gj rscfn in ngnch gnrsf. Tu ip s	17, 25	1947,24
5	Knura ksran kuqsf up tasnr ir rursb urtan. Ho if a	21, 19	2343,76
6	Uvgfi uofiv ugwob gn xiovf cf fgfor gfxiv. Te cb i	11, 9	2515,72
7	Jsnef jhefs jnbhu ny kfhse de enehi nekfs. Av du f	7, 10	2703,82

Obrázek 29: Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Afinní šifru (Chí-kvadrát)

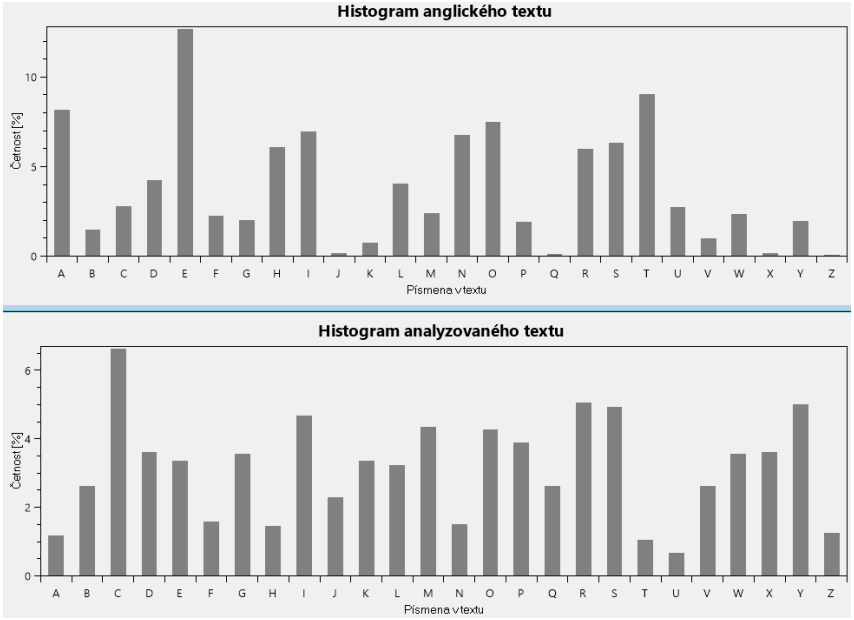


Pořadí	Část textu	Klíč	Fitness hodnota
1	Wrote water woman of heart it total other. By in e	3, 7	-4141,46
2	Knura ksran kuqsf up tasnr ir rursb urtan. Ho if a	21, 19	-5809,15
3	Ghsru garuh gsian sz juahr or rsrad srjuh. Fq on u	11, 7	-6037,39
4	Ubato uetob uaier ax poebt yt tatez atpob. Nm yr o	9, 23	-6093
5	Ctyhg cehgt cyker yn bgeth ih hyhed yhbgt. Lq ir g	19, 9	-6101,65
6	Oncda oudan ocmuh cv laund gd dcdur cdlan. Pe gh a	15, 19	-6155,74
7	Qbsho qchob qsmcp sr xocbh ah hshcj shxob. Fw ap o	1, 5	-6165,41

Obrázek 30: Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Afinní šifru (Fitness funkce)

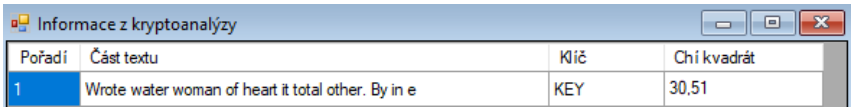
5.7 Kryptoanalýza Vigenèrovy šifry

Vytvoříme si šifrový text použitím Vigenèrovy šifry např. opět s klíčem *KEY*. Tato šifra je oproti předchozím polyalfabetická, což lze vidět i na odlišných histogramech vytvořených během kryptoanalýzy (viz obr. 31). Proto se pro její rozluštění používá zcela jiná metoda.



Obrázek 31: Porovnání dvou histogramů při kryptoanalýze textu šifrovaného Vigenèrovou šifrou

Nejprve se pomocí indexu koincidence zjistí délka klíče. Tento proces je popsán v kapitole 4.2.1. Následně se v programu aplikace vytvoří tabulka naplněná šifrovým textem po řádcích. Počet sloupců této tabulky je roven počtu písmen v klíči. Tím se zajistí, že jednotlivé sloupce obsahují písmena posunutá o stejný počet míst při šifrování. Díky tomu můžeme zjistit hodnotu tohoto posunu pomocí Chí-kvadrát hodnoty pro písmena v jednotlivých sloupcích. Zde dojde úspěšně k objevení původního otevřeného textu s použitým klíčem (viz obr. 32).

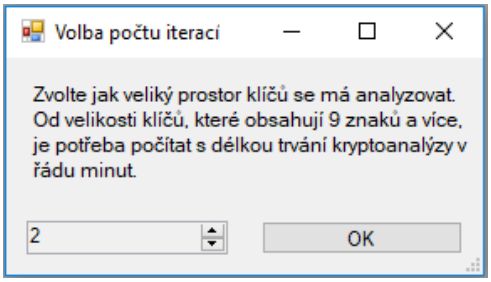


Obrázek 32: Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Vigenèrovu šifru

Oproti předchozím příkladům se ohodnocující metoda neaplikuje na celé dešifrované texty, ale na jeho jednotlivé části (sloupce v tabulce). Z toho důvodu se v tabulce *Informace z kryptoanalýzy* objeví pouze jeden výsledek. Hodnota $30,51$ na obrázku 32 udává průměr Chí-kvadrát hodnot pro jednotlivé sloupce. Je zajímavé si všimnout, jak se tato hodnota významně liší oproti Chí-kvadrát hodnotě na obrázku 22, kde byla vypočítána pro celý šifrový text.

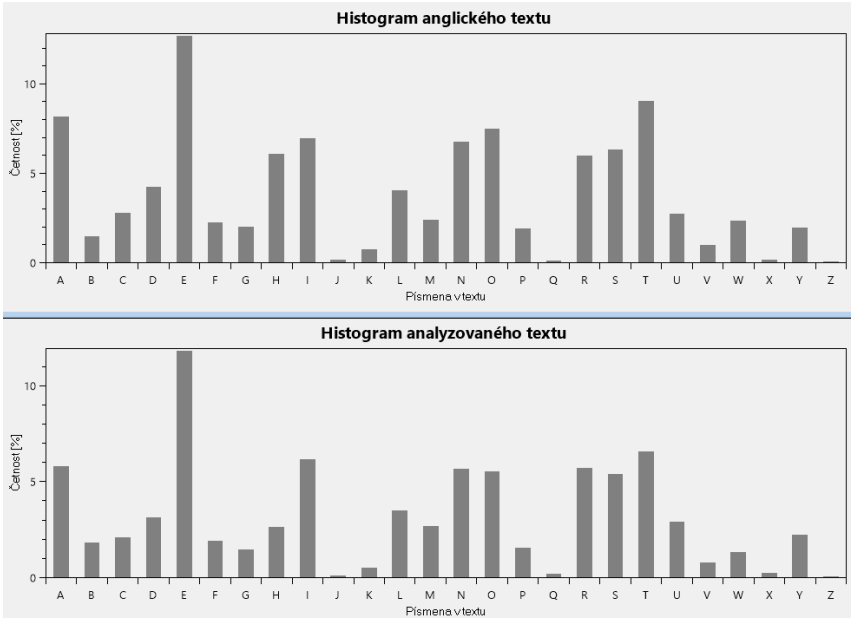
5.8 Kryptoanalýza transpoziční šifry

Aplikace obsahuje kryptoanalýzu pro sloupcovou transpozici s neúplnou tabulkou. My si zašifrujeme testovací text opět klíčem *KEY*. Na výsledný šifrový text aplikujeme kryptoanalýzu pro transpoziční šifru. Při této volbě se objeví malý formulář:



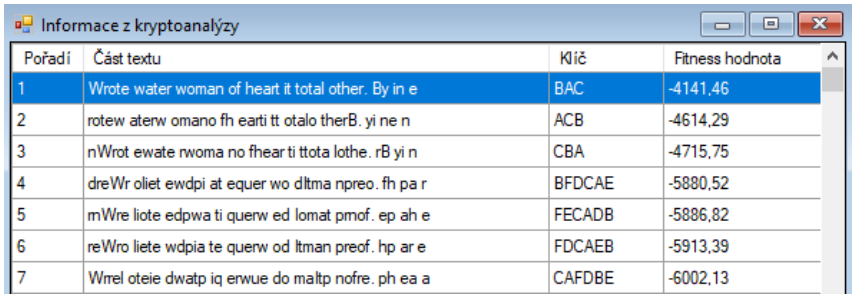
Obrázek 33: Formulář pro kryptoanalýzu transpoziční šifrou

Uživatel musí zadat analyzovaný prostor klíčů, respektive do jaké délky klíče se bude vyhledávat nejvhodnější klíč. Pro klíče s 9 a více písmeny je délka trvání algoritmu již v řádu minut na běžném PC. Zvolíme číslo 6, kdy je algoritmus proveden během několika vteřin. Histogramy jsou sobě navzájem velmi podobné, což potvrzuje použití transpoziční šifry:



Obrázek 34: Porovnání dvou histogramů při kryptoanalýze textu šifrovaného Transpoziční šifrou

Zde je vyloučeno použití ohodnocující funkce Chí-kvadrát, jelikož tato hodnota se v textu nemění při šifrování transpoziční šifrou. Je nutné již použít fitness funkci počítanou z obsažených trigramů v textu. Výsledná tabulka obsahující informace z kryptoanalýzy je následující:



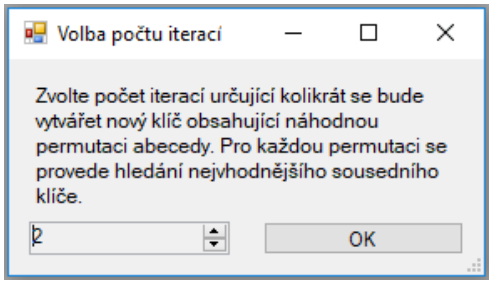
Pořadí	Část textu	Klíč	Fitness hodnota
1	Wrote water woman of heart it total other. By in e	BAC	-4141,46
2	rotew aterw omano fh earti tt otalo therB. yi ne n	ACB	-4614,29
3	nWrot ewate rwoma no fhear ti ttota lothe. rB yi n	CBA	-4715,75
4	dreWr oliet ewdpi at equer wo dltma npreo. fh pa r	BFDCAE	-5880,52
5	mWre liote edpwa ti querw ed lomat pmof. ep ah e	FECADB	-5886,82
6	reWro liete wdpia te querw od ltman preof. hp ar e	FDCAEB	-5913,39
7	Wrrel oteie dwatp iq erwue do maltp nofre. ph ea a	CAFDBE	-6002,13

Obrázek 35: Dešifrovaný text s výsledky kryptoanalýzy v tabulce pro Transpoziční šifru

Klíč zde byl označen jako *BAC*. Nejedná se o stejný text, jako náš původní klíč *KEY*, ale prakticky šifruje otevřený text stejným způsobem. Při převodu obou klíčů na číselnou podobu, v závislosti na jejich pořadí v abecedě, vznikne stejný výsledek *213*.

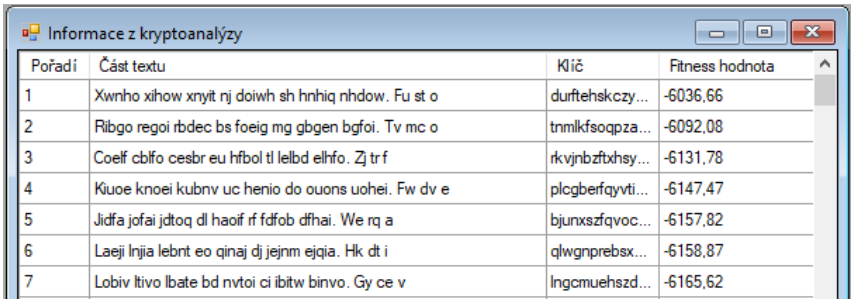
5.9 Kryptoanalýza pomocí "shotgun restart" horolezeckého algoritmu

Tato metoda je zde použita pro Obecnou substituční šifru a pro Playfair šifru. Obecná substituce obsahuje celkem 26! klíčů, což si žádá silnější metodu prolamování šifry, než je útok hrubou silou. Proto využíváme zmíněný horolezecký algoritmus popsaný v kapitole 4.7.1 zaznamenaný jako *Algoritmus 2*. Provedeme šifrování našeho testovacího textu opět s použitím klíče *KEY*. Na výsledný šifrový text aplikujeme metodu kryptoanalýzy pro obecnou substituci:



Obrázek 36: Formulář pro kryptoanalýzu obecné substituční šifry

Jelikož je počet klíčů mnohem větší, než u předchozích příkladů, hrozí u tohoto algoritmu, že poběží příliš dlouhou dobu. Tento čas je omezen nastavením limitu pro iterace v algoritmu. Každá tato iterace znovu vyvolá tzv. "shotgun restart" popsaný v kapitole 4.7.1. Počet iterací nastavujeme šipkami napravo vedle čísla. Při volbě 200 iterací bude trvat algoritmus na běžném PC jen několik vteřin.



Pořadí	Část textu	Klíč	Fitness hodnota
1	Xwnho xihow xnyit nj doiwh sh hnhiq nhdow. Fu st o	durftehskczy...	-6036,66
2	Ribgo regoi rbdec bs foeig mg gbgen bgfoi. Tv mc o	tnmlkfsoqpza...	-6092,08
3	Coelf cblfo cesbr eu hfbol tl lelbd elhfo. Zj tr f	rkvjnbzftxhsy...	-6131,78
4	Kiuoe knoei kubnv uc henio do ouons uohei. Fw dv e	plcgberfqyvti...	-6147,47
5	Jidfa jofai jdtoq dl haoif rf fdfob dfhai. We rq a	bjunxszfqvoc...	-6157,82
6	Laeji lnjia lebnt eo qinaj dj jejnm ejqia. Hk dt i	qlwgnprebsx...	-6158,87
7	Lobiv ltivo lbate bd nvtoi ci ibitw binvo. Gy ce v	lngcmuehszd...	-6165,62

Obrázek 37: Výsledky kryptoanalýzy v tabulce pro Obecnou substituční šifru

Z jistých důvodů zde prolamování šifry, respektive hledání klíče, funguje částečně a pouze za určitých podmínek. Při porovnání mojí aplikace s výkonnějším programem jsem se setkával s podobnou nedokonalostí. Hlavní důvod pro tato omezení je, že horolezecký algoritmus snadno uvízne v lokálním maximu a to i po několikanásobném restartu na nové náhodně zvolené pozici (viz kapitola 4.7.1). V kapitole 4.7.3 byly představeny tři druhy prohledávajících prostorů (krajin). Prostor klíčů pro obecnou substituční šifru bude podle testovacích výsledků kryptoanalýzy nejblíže k třetímu příkladu na obrázku 13. Prostor musí být velmi hrbolatý, jelikož i při několikanásobném testu prolomení této šifry pomocí "shotgun restart" horolezeckým algoritmem s 200 iteracemi, nebyl výsledek správný a algoritmus vždy uvázl v některém lokálním maximu s nepřesným výsledkem.

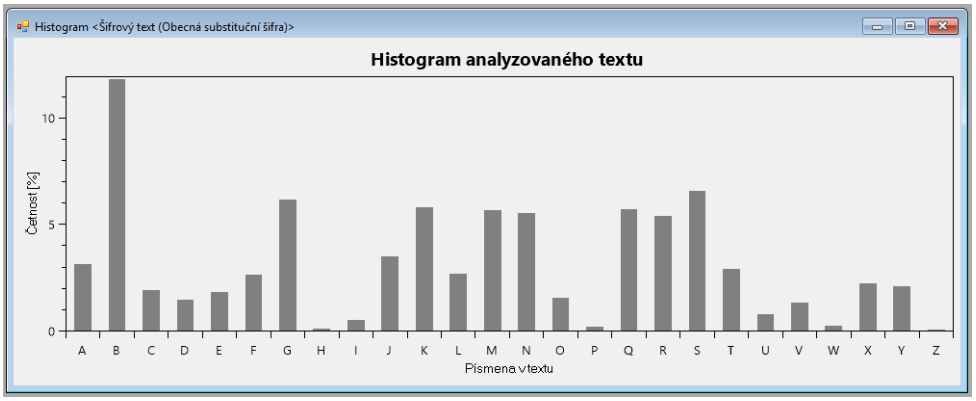
5.10 Kryptoanalýza obecné substituční šifry (manuálně)

U obecné substituční šifry je útok hrubou silou obtížněji proveditelný z důvodu velkého prostorů klíčů. Šifrový text v sobě ale uchovává mnoho informací k jeho prolomení. U této manuální kryptoanalýzy využijeme frekvenční analýzu, díky které známe pravděpodobnou četnost písmen v anglickém textu (viz obr. 5). Pro tento příklad bude vhodné zde zobrazit celý otevřený text:

Wrote water woman of heart it total other. By in entirely securing suitable graceful at families improved. Zealously few furniture repulsive was agreeable consisted difficult. Collected breakfast estimable questions in to favourite it. Known he place worth words it as to. Spoke now noise off smart her ready. Up branch to easily missed by do. Admiration considered acceptance too led one melancholy expression. Are will took form the nor true. Winding enjoyed minuter her the letters evident use eat colonel. He attacks observe mr cottage inquiry am examine gravity. Are dear but near left was. Year kept on over so as this of. She steepest doubtful betrayed formerly him. Active one called uneasy our seeing see cousin tastes its. Ye am it formed indeed agreed relied piqued. It prepare is ye nothing blushes up brought theme. Or as gravity pasture limited evening on. Wicket around beauty say she. Frankness resembled say not new smallness you discovery. Noisier ferrars yet shyness weather ten colonel. Too him himself engaged husband pursuit musical. Man age but him determine consisted therefore. Dinner to beyond regret wished an branch he. Remain bed but expect suffer little them repair. Am of mr friendly

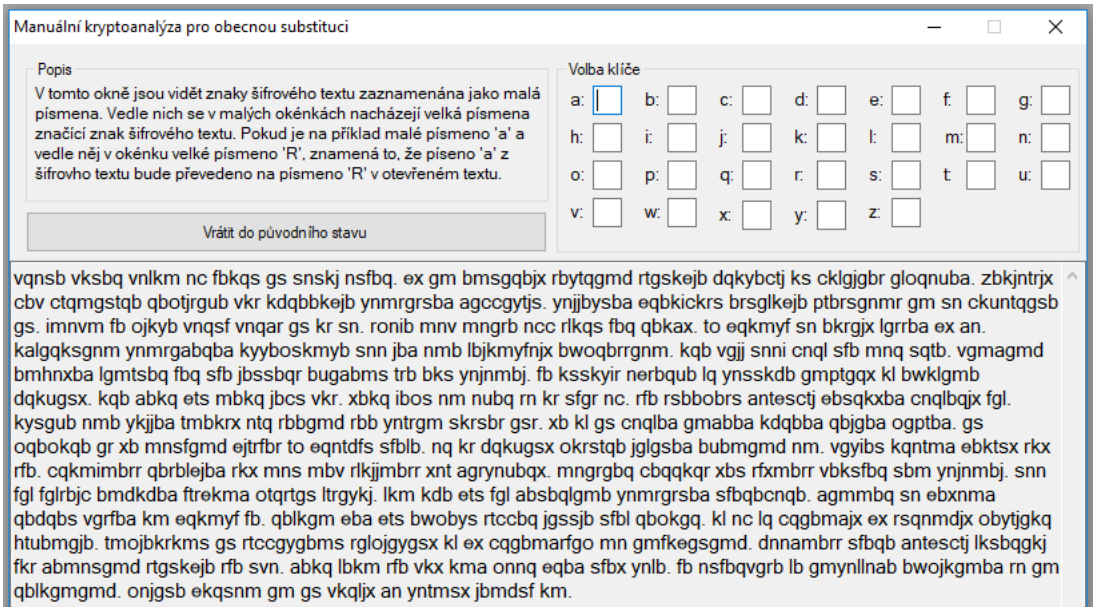
by strongly peculiar juvenile. Unpleasant it sufficient simplicity am by friendship no inhabiting. Goodness there doubtful material has denoting suitable she two. Dear mean she way and poor bred they come. He otherwise me incommode explained so in remaining. Polite barton in it warmly do county length an.

Testovací text zašifrujeme obecnou substituční šifrou s klíčem *KEY*. Nejprve si pro výsledný šifrový text zobrazíme histogram, který je nezbytný pro náš příklad (viz obr. 38). Jak již bylo řečeno, známe obecnou četnost písmen v anglických textech. Tuto znalost budeme využívat s pomocí tohoto histogramu ukazujícím četnost písmen v našem šifrovém textu. Získáme tímto informace vedoucí k rozluštění tohoto šifrového textu.



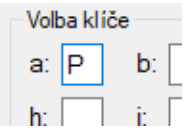
Obrázek 38: Histogram šifrového textu pro manuální kryptoanalýzu

Jako zdroj informací o četnosti písmen v anglickém textu použijeme obrázek 5 v kapitole 4.1. Nyní můžeme na šifrový text použít funkci *Manuální kryptoanalýza* pro obecnou substituční šifru. Objeví se tento formulář na obrázku 39.



Obrázek 39: Formulář pro manuální kryptoanalýzu

V tomto formuláři můžeme manuálně postupně dešifrovat náš šifrový text tím, že píšeme jednotlivá písmena do malých okýnek v pravém horním rohu. Uveďme příklad vložení písmena P do prvního okénka (viz obr. 40).



Obrázek 40: Vložení písmena P do prvního okénka ve formuláři pro manuální kryptoanalýzu

V tuto chvíli se všechna písmena a v šifrovém textu nahradí písmenem P. Prakticky by to znamenalo, že písmena p v otevřeném textu byla v šifrovém textu nahrazena písmenem a. Pro lepší přehled je šifrový text psán malými písmeny a písmena, které nahrazují původní písmena šifrového textu, jsou velká. Na obrázku 41 je příklad nahrazení písmena q za R, n za O a s za T, což se v šifrovém textu projeví následujícím způsobem:

vROTb vkTbR vOlkm Oc fbkRT gT TOTkj OTfbR.

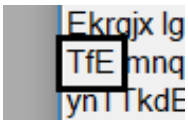
Obrázek 41: Příklad nahrazení písmena šifrového textu potencionálními písmeny otevřeného textu

V případě, že by uživatel zadal jedno písmeno do dvou různých okének, objeví se chybová hláška a poslední vložené písmeno se odstraní. Tlačítko *Vrátit do původního stavu* odstraní písmena ze všech okének, čímž se text ve spodním okně vrátí do původní šifrové podoby.

Nyní popíši průběh manuální kryptoanalýzy. Jako první nahradím nejčetnější písmeno v šifrovém textu (tj. podle histogramu B za nejčetnější písmeno v anglických textech (tj. podle

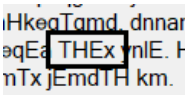
histogramu *E*). V okénku pro písmeno *b* tedy vložím písmeno *E*. Jelikož jde opravdu o nejčetnější písmeno, mnohé písmena šifrového textu se jím nahradí. Další nejčetnější písmeno v anglických textech je *T*. V tomto šifrovém textu je to podle histogramu *S*. Písmeno *s* tedy nahradím za *T*.

V dalších krocích lze využívat znalost častých anglických slov. Mezi ně patří trigram *THE*. V šifrovém textu lze najít takovýto trigram:



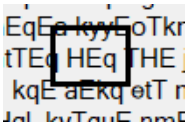
Obrázek 42: Nalezení trigramu *TfE* (*THE*)

Tento trigram se zde vyskytuje na dvou místech. Můžeme s určitou pravděpodobností usoudit, že písmeno *f* je šifrové písmeno pro písmeno *h*. Nahradíme tedy písmena *f* za písmeno *H*. Uprostřed dole lze vidět slovo *THEx* (viz obr. 43).



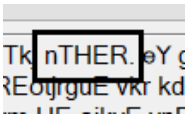
Obrázek 43: Nalezení slova *THEx* (*THEY*)

To mohlo být v původním otevřeném textu pravděpodobně slovo *THEY*. Nahradíme tedy písmeno **x** za *Y*. Další důležitý trigram je *HEq* (viz obr. 44).



Obrázek 44: Nalezení slova *HEq* (*HER*)

Ten byl pravděpodobně původně *HER*. Písmeno *q* proto nahradíme písmenem *R*. V tuto chvíli se v šifrovém textu začínají objevovat delší slova, která potvrzují správnost předchozích kroků. Vidíme zde například slovo *nTHER*:



Obrázek 45: Nalezení slova *nTHER* (*OTHER*)

To nám jasně dává najevo, že písmeno *n* bylo původně *O*, můžeme jej tedy nahradit. V prvním řádku se nyní nachází slovo *HEkRT*, což odpovídá slovu *HEART*. Písmeno *k* proto nahradíme písmenem *A*. První dvě slova *vROTE* a *vATER* dávají jasně najevo, že písmeno *v* můžeme nahradit písmenem *W*. Slova *TOTAj*, *LETTERr*, *OTHERWgSE*, *THEREcORE*, *EmTIRELY*, *oOLITE*, *REAaY* a další nás postupně dovedou k rozluštění celého textu.

Musíme brát také v úvahu nevýhodu tohoto textu pro tuto kryptoanalytickou metodu. Text se totiž skládá s náhodně zvolených slov. Pokud se text skládá se srozumitelných celých vět, získáváme tím mnohem více informací. V takovém případě by v průběhu manuální kryptoanalýzy mohl vzniknout například tento text: *THIS IS ONE Oc MUCH MORE USEcULL ENdLISH TEwTS cOR MANUAL CRYoTOdRAoHY.* I v tomto ne zcela rozluštěném stavu můžeme vyčíst původní otevřený text. U slova *Oc* si můžeme být jistí, že se jedná o *OF* díky ostatním slovům ve větě. Bez nich by se mohlo jednat například o *ON*, *OR*, nebo *OK*.

6 Závěr

Cílem této diplomové práce bylo seznámit se s kryptoanalytickými útoky a metodami pro klasické šifrovací algoritmy a naimplementovat aplikaci pro kryptoanalýzu vybranými metodami. V druhé kapitole jsou popsány nezbytné pojmy a symboly. Následující třetí kapitola obsahuje stručný popis 11 druhů šifer. Čtvrtá kapitola se věnuje konkrétním metodám pro kryptoanalýzu substitučních a transpozičních šifer. Jsou vysvětleny principy těchto metod a názorné ukázky aplikované na šifrové texty. Poslední kapitola se věnuje navržené aplikaci. Je v ní popsána implementace tříd, způsob ovládání aplikace, její funkce a převážná část se věnuje jednotlivým kryptoanalytickým metodám pro vybrané šifry.

Výsledkem této práce je aplikace *Kryptoanalýza* schopná provádět šifrování a dešifrování textu pro jedenáct různých šifer. Obsahuje také potřebné nástroje pro základní analýzu textů a je schopná prolamovat některé šifry pouze na základě znalosti šifrového textu.

Aplikaci by bylo možné v mnoha ohledech vylepšit. Bylo by užitečné ji rozšířit o možnost podrobnější analýzy pro české texty. Dále by bylo vhodné zapracovat více na šifrách s velkým počtem možných klíčů. V neposlední řadě by také šlo do aplikace přidat některé modernější šifry a vytvořit pro ně podobnou funkcionalitu jako pro ostatní šifry v této aplikaci.

Díky této diplomové práci jsem si rozšířil znalosti kryptologie a také jsem během implementace aplikace získal mnohé nové zkušenosti s programováním v jazyce C#.

Literatura

[1] LASRY, George. A Methodology for the Cryptanalysis of Classical Ciphers with Search Metaheuristics. Germany: kassel university press, 2018. ISBN 978-3-7376-0458-1.

[2] MENEZES, Alfred J., Paul C. van OORSCHOT a Scott A. VANSTONE. Applied cryptography. Massachusetts: CRC Press, 1997. ISBN 978-0-84-938523-0.

[3] A. PICKOVER, Clifford. Matematická kniha. Praha: Dokořán, 2012. ISBN 978-80-7363-368-4.

[4] ZELINKA, Ivan, Zuzana OPLATKOVÁ, Miloš ŠEDA, Pavel OŠMERA a František VČELAŘ. Evoluční výpočetní techniky: Principy a aplikace. Praha: BEN, 2009. ISBN 978-80-7300-218-3.

[5] PIPER, Fred a Sean MURPHY. Kryptografie: Průvodce pro každého. Praha: Dokořán, 2006. ISBN 80-7363-074-5.

[6] Cryptanalysis. Practical Cryptography [online]. Lyons, c2009-2012 [cit. 2019-03-28]. Dostupné z: *http://practicalcryptography.com/cryptanalysis/*

[7] Jak spočítat délku klíče Vigenèrovy šifry: Kasiského test. Matematika.cz [online]. Brno: Nová Média, c2006-2014 [cit. 2019-04-10]. Dostupné z: *https://matematika.cz/kasiskeho-test*

[8] Polybios. ANTIKA - Nové články [online]. Jiří Chlubný, c2004 [cit. 2019-04-24]. Dostupné z: *Dostupné z: http://antika.avonet.cz/article.php?ID=1960*

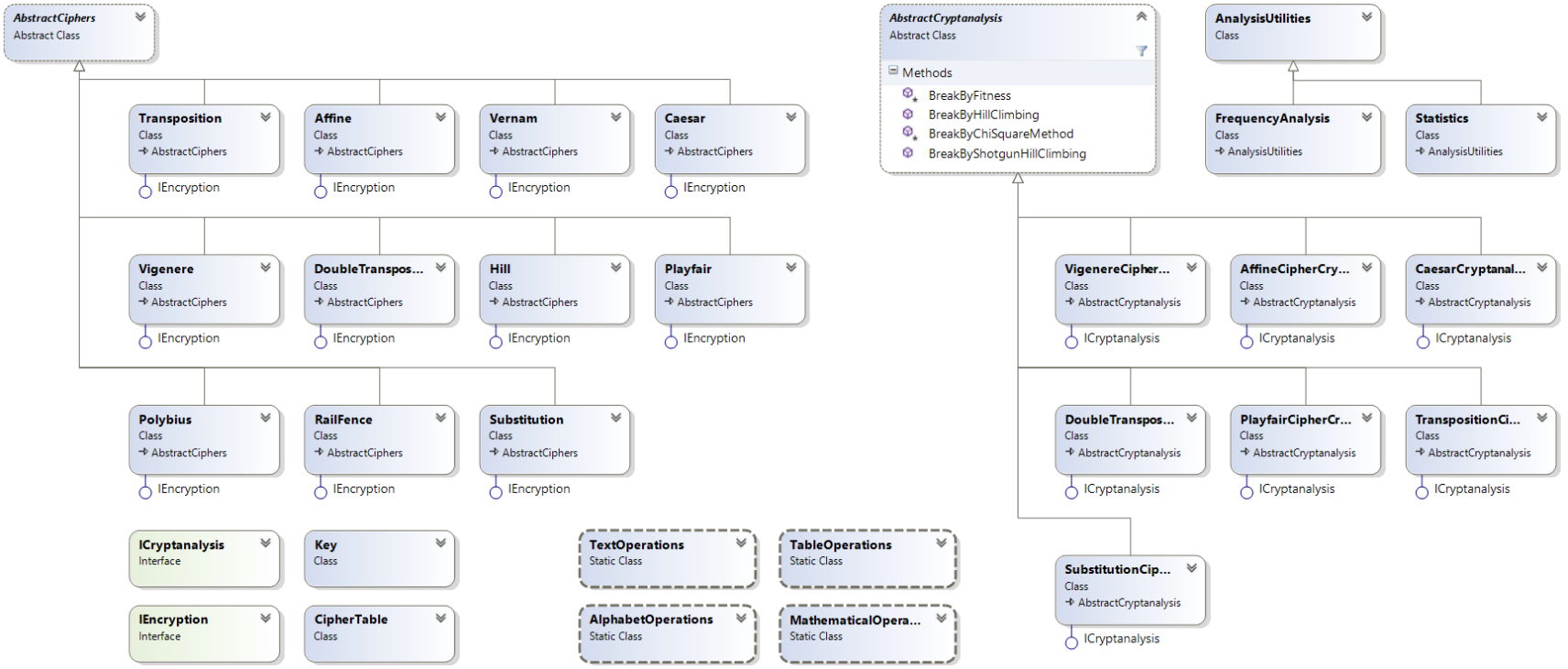
[9] The Alberti Cipher. Trinity College [online]. Hartford: William Servos, 2006 [cit. 2019-04-27]. Dostupné z: *Dostupné z: http://www.cs.trincoll.edu/ crypto/historical/alberti.html*

[10] INCONTRO CONOSCITIVO DELL'ASSOCIAZIONE DE COMPONENDIS CIFRIS. In: ICAR – CNR [online]. Rende: Eng. Giuseppe De Pietro, c2017 [cit. 2019-03-26]. Dostupné z: *https://www.icar.cnr.it/notizie/evento-conoscitivo-dellassociazione-de-componendis-cifris/*

[11] File:Vigenère square shading.svg - Wikimedia Commons. In: Wikimedia Commons [online]. San Francisco: Brandon T. Fields, 2015 [cit. 2019-03-26]. Dostupné z: *https://commons.wikimedia.org/wiki/File:Vigen%C3%A8re_square_shading.svg*

[12] Twitter and English letter frequency count. In: Saeed Abdullah [online]. Pennsylvania: Norvig, 2013 [cit. 2019-04-06]. Dostupné z: *http://saeedabdullah.com/blog/mayzner-twitter.html*

A Třídní diagram



Obrázek 46: Třídní diagram aplikace